Access Control in IoT/M2M - Cloud Platform

Anggorojati, Bayu

Publication date:
2015

Document Version
Accepted author manuscript, peer reviewed version

Access Control in IoT/M2M - Cloud Platform

Bayu Anggorojati
Center for TeleInFrastruktur

A DISSERTATION SUBMITTED TO
THE DEPARTMENT OF ELECTRONIC SYSTEMS OF AALBORG UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Supervisor:
Associate Professor Neeli Rashmi Prasad, Aalborg University, Denmark
Professor Ramjee Prasad, Aalborg University, Denmark

The examination committee:
Professor Josef Noll
Professor Milica Pejanovic-Djurisic
Associate Professor Zheng-Hua Tan (Chairman)

Moderator:
Associate Professor Albena Mihovska

Date of defence: 25 February 2015

ISSN: *** - ****
ISBN: 978-87-7152-064-4

Mandatory page

Thesis title: Access Control in IoT/M2M - Cloud Platform

Name of PhD student: Bayu Anggorojati

Name and title of supervisors:

- Associate Professor Neeli Rashmi Prasad
- Professor Ramjee Prasad

List of published papers:

- **B. Anggorojati**, N.R. Prasad, R. Prasad, "Efficient Fine Grained Access Control for RFID Inter-Enterprise System," Journal of Cyber Security and Mobility, vol. 2, no. 3 & 4, pp. 221-242, 2013
- P.N. Mahalle, **B. Anggorojati**, N.R. Prasad, R. Prasad, "Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things," Journal of Cyber Security and Mobility, vol. 1, no. 4, pp. 309-348, 2013.
- **B. Anggorojati**, P.N. Mahalle, N.R. Prasad, R. Prasad, "Secure Access Control and Authority Delegation based on Capability and Context Awareness for Federated IoT," Internet of Things and M2M Communications. ed. / Fabrice Theoleyre; Ai-Chun Pang. Denmark : River Publisher, 2013. p. 135-160 (The River Publishers Series in Information Science and Technology).
- **B. Anggorojati**, N.R. Prasad, R. Prasad, "Secure Capability-based Access Control in the M2M Local Cloud Platform," Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2014 IEEE 4th International Conference on, May, 2014.

- **B. Anggorojati**, N.R. Prasad, R. Prasad, "An Intrusion Detection Game in Access Control System for the M2M Local Cloud Platform," the 19th Asia Pasific Conference on Communications (APCC) 2013, Bali, Indonesia, 2013.
- **B. Anggorojati**, P.N. Mahalle, N.R. Prasad, R. Prasad, "Efficient and Scalable Location and Mobility Management of EPCglobal RFID System," the 16th WPMC, Atalantic City, NJ, USA, 2013.
- **B. Anggorojati**, P.N. Mahalle, N.R. Prasad, R. Prasad, "Capability-based Access Control Delegation Model on the Federated IoT Network," the 15th WPMC, Taipei, Taiwan, 2012.
- P.N. Mahalle, **B. Anggorojati**, N.R. Prasad, R. Prasad, "Identity driven Capability based Access Control (ICAC) scheme for the Internet of Things," the 6th IEEE ANTS, Bengaluru, India, 2012.
- P.N. Mahalle, **B. Anggorojati**, N.R. Prasad, R. Prasad, "Identity Establishment and Capability Based Access Control (IECAC) Scheme for Internet of Things," the 15th WPMC, Taipei, Taiwan, 2012.

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Abstract

Billions of devices are connected to the Internet nowadays, and the number will continue to grow in the future thanks to the advances in the electronics and telecommunication technology developments. Its application in broad aspects of human's life brings a lot of benefits by improving productivity and quality of life. This paradigm, which is often called Internet of Things (IoT) or Machine-to-Machine (M2M), will provide an unprecedented opportunity to create applications and services that go far beyond the mere purpose of each participant.

Many studies on the both technical and social aspects of IoT have shown that the concern about the security and privacy play a huge role for the mass adoption of the IoT/M2M as cloud services. Among the important topics within the security and privacy, the access control is an important mechanism, which essentially manages how the important assets or resource of a system can be accessed by other parties by means of a set of access policies.

For an IoT system such as Radio Frequency Identification (RFID) that collects huge amounts of RFID events data and may store it in the cloud storage for tracking purpose, access control to such data becomes a critical point to the privacy of the enterprises as well as the customers. Certainly, designing an access control to the RFID events data with high-granularity is desirable to maintain the privacy while allowing external party to perform tracking and tracing of RFID tags. In addition, mobility or location management also plays a big role to perform tracking of RFID tags. Scalability and efficiency are two important requirements in location management when big numbers of tags are moving from one reading location to the others, i.e. being mobile. Thus, designing a fine-grained access control along with scalable location management in RFID system is of paramount importance.

A distributed cloud platform approach for the IoT/M2M, which consists of a set of IoT/M2M gateways, is introduced to cope with some inherent issues of IoT network which is highly heterogeneous and distributed in nature. As a

result, access control becomes even more challenging when such approach -also called as local cloud - which may consist of devices with low computational capacity, is used. As each of the IoT/M2M gateways may have different assets or resources, and thus different access policies, combining different policies and to make an access control decision in distributed manner is a very challenging task. In addition, the access control system should also fulfill other requirements in terms of scalability, context-awareness, flexibility, and attack resilience. These challenges lead us to come up with capability-based access control that can be easily distributed, i.e. scalable and suitable for distributed system, and propagated, i.e. allow flexible access delegation. On top of that, contextual information can also be included in the capability data structure so as to deal with dynamic context in IoT/M2M environment. However, thorough design of capability-based access control is needed, especially to keep the access delegation through capability propagation under control and to maintain secure access control.

To detect and mitigate various threats, especially the insider threat, within the IoT/M2M local cloud platform is a difficult task for the access control system. Thus, an Intrusion Detection System (IDS) is needed as the integral part of the access control system. We can imagine a situation where a malicious node disguises as a good node such that it can join the local cloud, but once it becomes part of the cloud it would cause a huge damage to the system. For example it could manipulate access right of an actuator controlled by a gateway, e.g. to open a gate or turning on or off some switches, stealing some sensitive data from sensors, and so on. Keeping in mind such threat and the fact that minimum human interaction is needed in the local cloud environment, the IDS should be able to learn and update its knowledge based on the interaction with the other nodes. This leads us to study, model, and analyze the interactions between malicious node and regular node equipped with IDS with game theory, in order to suggest the best strategies for both sides. The study also includes a general fact that each node has a set of assets or resources with different values. Finally, an optimum strategy for both attacker and defender will be derived by considering their respective costs and benefits.

Dansk Resume

Milliarder af apparater er forbundet til internettet i disse dage og antal af disse apparater vil stige i fremtiden takket være udviklingen i elektronik og telekommunikation. Dens anvendelse i flere aspekter af menneskeliv har bragt en masse fordele ved at forbedre produktivitet og livskvalitet. Dette paradigme, som ofte hedder Internet of Things (IoT) eller Machine-to-Machine (M2M), vil give en hidtil uset mulighed for at skabe applikationer og tjenester, som strækker langt over det behov, som hver forbruger har.

En del studier på både de tekniske og sociale aspekter af IoT har indikeret, at bekymringen for sikkerhed og privatliv spiller en stor rolle for anvendelse af IoT/M2M som cloud services. Udover sikkerhed og privatliv er adgangskontrol også en vigtig mekanisme, som essentielt styrer vigtigheden af andre partier til at få adgang til et systems aktiv eller ressourcer ved hjælp af et sæt af adgangspolitikker. For et IoT system såsom Radio Frequency Identification (RFID), som samler store mængder RFID events data og som kan opbevares i cloud storage således for at kunne bliver sporet, adgangskontrol til dataene bliver et vigtigt punkt for virksomhedernes og kundernes privatliv. At designe en adgangskontrol til RFID events data med en høj nøjagtighed er bestemt ønsket for at kunne opretholde privatlivet og samtidig tillade eksterne partier at foretage sporingen af RFID tags. Ydermere spiller mobilitets og beliggenhed styring også en stor rolle i at foretage sporing af RFID tags. Skalerbarhed og effektivitet er de to vigtige krav i styringen af beliggenheden når store mængder af data bevæger sig fra en læsebeliggenhed til andre beliggenheder ved eksempelvis at være bevægelig. Derfor er det vigtigt at designe en finkornet adgangskontrol sammen med en skalerbar beliggenhedstyring i RFID systemet.

En distribueret skyplatform tilgang til IoT/M2M, som består af et sæt IoT/M2M gateways, er blevet introduceret for at kunne håndtere nogle iboende sager vedr. IoT netværk, som i høj grad er forskellige og distribueret af karakter. Dette medfører, at adgangskontrollen bliver mere udfordrende når der anvendes sådan en tilgang, som også hedder local cloud og som kan bestå af

apparater med en lav beregningsmæssig kapacitet. Da hver IoT/M2M gateway kunne have forskellige aktiver eller ressourcer og dermed ogsåforskellige adgangspolitikker, at kombinere diverse politikker og foretage en beslutning om håndteringen af adgangskontrollen er en meget krævende opgave. Desuden skal adgangskontrolsystemet også opfylde andre krav med henblik på skalerbarhed, kontekstbevidsthed, fleksibilitet og angrebsmodstandskraft. Disse udfordringer leder frem til udviklingen af en kapacitetsbaseret adgangskontrol, som let kan distribueres ved fx at være skalerbar og egnet til et distribueret og spredt system. Et distribueret og spredt system angiver hermed en fleksibel adgangsuddelegering. Oven i købet vil kontekstuelle data også kunne være inkluderet i kapacitetsdatastrukturen for at kunne håndtere den dynamiske kontekst i IoT/M2M miljøet. Det skal dog bemærkes, at der skal et grundigt design af en kapacitetsbaseret adgangskontrol til for at styre adgangsuddelegeringen gennem kapacitetspredningen og for at opretholde en sikker adgangskontrol.

At detektere og afbøde forskellige trusler, især interne trusler i IoT/M2M lokal skyplatform, er en besværlig opgave for adgangskontrolsystemet. Derfor er det nødvendigt at have en Intrusion Detection System (IDS) som en central del af adgangskontrolsystemet. Man kan forestille sig en situation, hvor en farlig node skjuler sin identitet og lader som om, den er en ufarlig node. Når truslen trænger ind i den lokale sky kan den faktisk medføre en stor og alvorlig skade på systemet. For eksempel kan den manipulere en aktuators adgangsret, som er styret af en gateway ved at lukke op for indgangen i systemet eller slukke nogle knapper i systemet, at stjæle nogle ømtålelige data fra sensorerne, osv. Af hensyn til sådanne trusler og nødvendigheden af minimum menneskeinteraktionen i det lokale skymiljøburde IDS kunne lære og opdatere dets viden på baggrund af interaktionen med andre noder. Dette fører frem til at studere, analysere og modellere interaktioner mellem farlige og almindelige noder forsynet med IDS med spilteori for at kunne anbefale de bedste strategier for begge parter. Studiet omfatter også faktummet, at hver node har en række aktiver eller ressourcer med forskellige værdier. Endelig vil en optimal strategi for både angriber og forsvarer kunne opnås ved at tage højde for deres respektive omkostninger og fordele.

Table of Contents

List of Figures

List of Tables

List of Acronyms

FI Future Internet

EC European Commission

M2M Machine-to-Machine

ITS Intelligent Transport System

IETF Internet Engineering Task Force

MANET Mobile Ad-hoc NETwork

CCAAC Capability-based Context-Aware Access Control

IoT Internet of Things

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

WSN Wireless Sensor Network

NFC Near Field Communication

ICAP Identity based Capability

VID Virtual Identity

MAGNET My Adaptive Global NETwork

CASM Context Aware Security Manager

PN Personal Network

ACS Access Control Servers

ACL Access Control List

ACM Access Control Matrix

RBAC Role Based Access Control

ABAC Attribute Based Access Control

XACML Extensible Access Control Markup Language

GTRBAC General Temporal RBAC

TRBAC Temporal RBAC

CARBAC Context Aware Role Based Access Control

XML Extensible Markup Language

PTD Personal Trusted Device

WAN Wide Area Network

WWAN Wireless Wide Area Network

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

PAN Personal Area Network

CA Certificate Authority

PN-F Personal Network Federation

AVISPA Automated Validation of Internet Security Protocols and Applications

IdP Identity Provider

SDP Security Decision Point

6LoWPAN IPv6 over Low-power Wireless Personal Area Network

LoWPAN Low-power Wireless Personal Area Network

DHT Distributed Hash Table

P2P Peer-to-Peer

RP Reader Protocol

LLRP Low Level Reader Protocol

ALE Application Level Events

HAL Hardware Abstraction Layer

FC Filtering and Collection

BEG Business Event Generator

EPCIS Electronic Product Code Information System

ONS Object Name Server

NAPTR Naming Authority PoinTeR

TDS Tag Data Standard

TDT Tag Data Translation

JAXB Java Architecture for XML Binding

SOAP Simple Object Access Protocol

RDBMS Relational Data Base Management System

AAL Auto-ID Authorization Language

SQL Structured Query Language

SIP Session Initiation Protocol

VoIP Voice over IP

IMS IP Multimedia Subsystem

SUA Surrogate User Agent

SIMPLE SIP for Instant Messaging and Presence Leveraging Extension

SNS SRMS Name Server

SRV SeRVice

RF Radio Frequency

EPC Electronic Product Code

ICT Information and Communication Technology

IoT Internet of Things

NFC Near Field Communication

RFID Radio Frequency IDentification

XML eXtensible Markup Language

DNS Domain Name System

URI Uniform Resource Identifier

IP Internet Protocol

NGN Next Generation Network

JSON Java Script Object Notation

GPS Global Positioning System

CDF Cummulative Distribution Function

API Application Programming Interface

PKI Public Key Infrastructure

ECC Elliptic Curve Cryptography

ECMQV Elliptic Curve Menezes-Qu-Vanstone

SaaS Software as a Service

PaaS Platform as a Service

IaaS Infrastructure as a Service

TaaS Thing as a Service

ASPIRE Advance Sensors and Programmable middleware for Innovative Rfid Enterprise applications

ISISEMD Intelligent System for Independent living and SElfcare of seniors with cognitive problems or Mild Dementia

BETaaS Building the Environment for Thing as a Service

IDS Intrusion Detection System

PBE Perfect Bayesian Equilibrium

NE Nash Equilibrium

CoAP Constrained Application Protocol

MQTT Message Queuing Telemetry Transport

CRM Customer Relationship Management

NIST National Institute of Standards and Technology

SEC Standard of Efficient Cryptography

1 Introduction

The goal of this chapter is to motivate the necessities in securing, as well as ensuring the privacy and trust, of the Internet of Things (IoT)/Machine-to-Machine (M2M) and Cloud, by taking into consideration the huge demands of such applications nowadays and in the future. The glimpse of the key issues is taken in this chapter which are identified and addressed in the dissertation. It includes explanation of contributions of the dissertation and related publications which are provided. Finally, outline of the thesis is specified that gives an overview of the individual chapters.

1.1 Motivation

Billions of devices are connected to the internet nowadays, and the number will continue to grow in the future. As documented in [37] back in 2004, Harbor Research, a technology consultancy and analysis firm, predicted that by 2010, at least 1.5 billion devices will be Internet-connected worldwide. Some advances in the electronics and telecommunication technology developments in recent years, especially in wireless communication, that result in various kinds of tiny yet powerful devices with communication and networking capabilities, have attracted the industries to adopt this technology into their everyday business, so as to increase their efficiency, i.e. saving time, money or both [14].

Moreover, there is also a huge demands in other sector other than industrial sector, such as public service, assisted living, etc, for this Information and Communication Technology (ICT) developments.

Therefore, this calls the needs to a new paradigm in machine-to-machine communication which enables object or "things" connectivity to the global internet network. This paradigm has been "popularized" in the recent years by the term IoT or M2M. Before going even further, it is important to note that the term IoT and M2M are often referring to the same thing or used interchangeably. One of the articles found in the literature that use both of the term Internet of Things and M2M is [14]. The article was entitled "The Internet of Things", and the author of this article said that M2M communications would bring about a new wave of growth in the global economy driven by billions of electronic and electromechanical devices being connected to the Internet. In the recent years, the terms M2M/IoT have been used together in many scientific conferences within the field of Information and Communication Technology.

The vision of connecting billions and even trillions of devices to the internet have come closer to the realization nowadays, driven by many kinds of applications that not only concern with the industries but also the government as well as public institution, and even everyday people's life. For instance, smart grid and smart metering allow the electrical utility companies to monitor the supply of electrical power by power plants, including those that are owned by private house, especially the ones with renewable energy sources, and the energy demands by means of monitoring devices, e.g. wireless sensors. Moreover, the electricity prices that are based on supply and demand can also be monitored by the users, so as they can utilize their electrical appliances when the electricity price goes low due to high supply and so on. Another example is the Intelligent Transport System (ITS) application in which information about the traffic in a city can be monitored by wireless sensors or video surveillance, and then communicated to the users on their mobile devices with the help of a Global Positioning System (GPS) transceiver, so that users will get the information about the best route by avoiding traffic jam, or the road accident can be prevented due to bad road condition, bad weather, natural disaster, e.g. avalanche, landslide, or flood, and so on. In fact, there are a lot more examples and applications in different areas that can utilize the M2M or IoT, such as e-Health, smart home and building, environment monitoring, etc. All those examples show that massive amount of data generated by billions of devices will be transferred through the network, and eventually the internet. In one hand, it opens a completely new opportunities and business models, especially to telecommunication industries due to high amount of generated data traffic

as well as to the ICT players in general, and also growth in the global economy in general. On the other hand, it leaves some problems and challenges to be tackled, in particular the security and privacy issues.

Another statement that can support the previous paragraph can be found in [15], in which US National Intelligence Council (NIC) in its Conference Report on Disruptive Civil Technologies has included IoT as one of "Six Technologies with Potential Impacts on US Interests out to 2025". It was foreseen that the internet nodes will reside in almost all everyday things such as food packages, furniture, paper documents, and more, and therefore, it could contribute invaluably to the economy. Besides the potential opportunities of IoT to the US economy, [15] also stressed out the potential risk of IoT which is the loss due to the security issue that could be exploited from IoT. This really shows how huge the impact of IoT to the economy and people's everyday life is, and also the risk it might cause to the security, not to mention the loss in privacy due to security breach of people's personal private information communicated over those devices.

Smart things or objects connected to the Internet can take advantage of services that are offered by the cloud [35]. One obvious benefit of integrating IoT/M2M with cloud is to give flexibility to the user in accessing services offered by the M2M application through web interface. Thus, it also gives flexibility to the M2M service provider to offer its services to even more users or customers. Besides that, utilizing cloud services, thanks to the advances in the cloud computing technology, could bring more advantages to the actual deployment of M2M or IoT. First of all, an IoT device could save the energy consumption, to a certain extent, by offloading its computation to the cloud infrastructure. That is, if the computational energy required to perform a certain task locally is higher then sending the data to the cloud through the internet [36]. Nevertheless, this kind of solution faces another unique challenges, such as security and privacy, communication reliability, and real-time data requirement in some IoT/M2M applications [36]. Secondly, centralized decision making and management of billions of devices within the cloud will become an essential value of the IoT vision, although decentralized processing is critical to address the complexity of M2M applications [60].

Cloud computing itself is becoming a popular paradigm in the internet computing world, that provides a model to support processing large volumetric data using clusters of commodity computers [50]. This computing model allows on-demand, pay-for-use, and more economical, scalable IT services, over the Internet. Moreover, it provides many advantages for businesses—including low initial capital investment, shorter start-up time for new services, lower

maintenance and operation costs, higher utilization through virtualization, and easier disaster recovery—that make cloud computing an attractive option [36]. Cloud platforms are dynamically built through virtualization with provisioned hardware, software, networks, and datasets [24]. Thanks to the virtualization technology, the computing at the local devices, such as PC, mobile and M2M devices, can be migrated to a virtual server clusters at the data center. However, the lack of trust on the security and privacy protection of cloud users information to the cloud providers prevents the wide scale acceptance of cloud based computing services, which call the needs of designing a secure, trustworthy, and dependable cloud platform.

1.2 IoT/M2M and Cloud Service: An Introduction

In the field of ICT, there is no single universally accepted definition of the term IoT or M2M, although it is used to refer to a vision of connecting multi-billions of electronics and communication devices, including all everyday things powered by "smart" objects, to the internet. The vision of IoT is very much influenced by the term "Internet of Things" itself, which syntactically is composed of two terms: "Internet" and "Things" [9]. As a result, the visions one IoT paradigm is divided into two main visions, namely the network oriented vision of IoT or the "Internet oriented" and the "Things oriented" vision which focus on generic "objects" to be integrated into a common framework.

The very first definition of IoT derives from a "Things oriented" perspective; the considered things were very simple items: Radio Frequency IDentification (RFID) tags. The terms "Internet of Things" is, in fact, was popularized by the work of the Auto-ID Center at the Massachusetts Institute of Technology (MIT), when its first white paper [54] already suggested a vision that extended beyond RFID [43]. Auto-ID center together with EPCGlobal have been focusing mainly on the development of the Electronic Product Code (EPC) to support a widely use of the RFID technology in the modern trading and supply chain networks worldwide, and to create the industry-driven global standards for the EPCglobal Network. These standards are primarily designed to improve object traceability and the awareness of its status, current location, etc. This is without doubt a key component towards the full deployment of the IoT vision. However, it is not the only one.

An article in [48] stated that RFID still stands at the forefront of the technologies driving the vision due to the RFID maturity, low cost, and strong

support from the business community. However, the authors in [48] also stated a wide portfolio of device, network, and service technologies, together with RFID, will eventually build up the IoT, such as Near Field Communication (NFC), WSAN (Wireless Sensor and Actuator Network). An example of a combined RFID and wireless sensor network is WISP (Wireless Identification and Sensing Platforms) [49] in which a sensing platform is developed on top of passive RFID tags based on EPCGlobal standards, in particular EPC Class 1 Generation 1 and 2 standards, which operate in the UHF bands.

On the other hand, the term M2M was initially introduced by carrying the "Things" oriented vision, driven by the industries to help them in increasing their efficiency in the industrial process. M2M telemetry made a breakthrough already in the early 2000s, with the use of cellular modems and IP to monitor and control a wide range of equipment from vending machines to water pumps [55]. The "hype" towards M2M communication was further elevated along with the developments of some new standards in wireless communication utilizing the free-unlicensed ISM band in the early 2000, such as the IEEE 802.11 that is known as Wireless Local Area Network (WLAN) standard; IEEE 802.15.1 or Wireless Personal Area Network (WPAN) standard that was ratified in the early version of Bluetooth technology; and IEEE 802.15.4, which is a low rate WPAN used for short range communication, and is adopted by ZigBee. The implementation of those wireless standards have been widely used since then by the industries in different sectors, ranging from remote monitoring and measurements, to the industry automation and robotics. Earlier, some companies that are already deploying M2M system based on those wireless standards or other technologies, would rather call their applications as either "building automation", "patient monitoring", "automated meter reading", "automated asset tracking", and so on, instead of the M2M technology [14]. Nowadays, M2M applications are becoming broader, touching almost any aspect of people's life and any communication-capable devices. The authors in [60] envisioned the future M2M towards the embedded internet. This vision allude to embedding more intelligence and functionality, such as security-on-chip, system analytic, embedded augmented sensing, and plug-and-play capability, to the end device.

On the networking oriented category falls the IoT vision of the IPSO (IP for Smart Objects) Alliance. This forum suggests Internet Protocol (IP) to be the main networking protocol towards interconnecting billions of communication-capable smart "things" or objects [16]. The main reasons for pushing IP for IoT are due to its maturity, interoperability, scalability, manageability, and the fact that it has been widely accepted network and communication society. Furthermore, in some of IPSO's whitepapers [1, 23], it was shown that through a

wise IP adaptation and by incorporating IEEE 802.15.4 into the IPv6 architecture, i.e. IPv6 over Low-power Wireless Personal Area Network (6LoWPAN), the IP stacks consume a small memory footprint. This development confutes the belief that IP is such a heavyweight protocol. Hence, the full deployment of the IoT paradigm will be enabled by using IP as the networking protocol. Internet-0 or "Internet-Zero" follows a similar approach of reducing the complexity of the IP stack to achieve a protocol designed to interconnect all type of devices over IP, a concept known as interdevice internetworking [19]. According to both the IPSO and Internet-0 approaches, the IoT will be deployed by means of a sort of simplification of the current IP to adapt it to any object and make those objects addressable and reachable from any location [9].

Another work related to the networking oriented of IoT vision was done in the context of MAGNET and MAGNET Beyond projects [47]. MAGNET's primary focus was on the concepts of Personal Network (PN) and its extension, Personal Network Federation (PN-F), enabling wireless machine-to-machine communication literally, from within the short range communication, e.g. WPAN, and even goes across different geographical positions. Among others, its networking protocols, including the PN and PN-F creation as well as the interoperability with the existing network, such as 3G and beyond 3G; and a solid security framework which includes identity, trust, and key management; have been some key contributions towards opening path of the IoT vision.

1.3 State of the Art: IoT/M2M Cloud Platform and the Enabling Technologies

This section gives an overview of the related theories as well as the state of the art of technologies and research and development works, which are important to this PhD work. It will begin with the description of the enabling technologies within IoT/M2M, including the RFID as an identification, sensing, and communication technology; IoT/M2M middleware that bridges the physical world with the application layer; and the network architecture, enabling the actual machine-to-machine communication through internet. It will then followed by some brief overview of cloud computing technology and its taxonomy. Finally the state-of-the-art of the IoT/M2M and cloud platform, and some research works on the security, privacy, and trust in such platform will be presented.

1.3.1 IoT/M2M Physical Sensing: RFID as an Identification, Sensing, and Communication Technology

RFID is one of the most promising identification, sensing as well as communication technology to enable the actual deployment of IoT. One of its important feature that requires no power source, i.e. passive RFID, and moreover its price that decreases over time, along with the invention of new technologies and efficient RFID production, have become the main reason for industries to adapt it into their business process recently. Furthermore, the development of sensing platform on top of passive RFID tag, such as the WISP project by Intel [49], ALB-2484 by Alien Technology, TELID 210 by Microsensys, and [25] that is developed by three Japanese companies [45], is giving more degree of freedom to the RFID to be used in a remote sensing type of application by the industries, e.g. the cold supply chain. In addition to that, some research works and commercial products attempted at integrating the RFID system with the Wireless Sensor Network (WSN), either at the network level [13], or tag level, i.e. integrated RFID tag and sensor node, [52, 45], enable the multi-hop communication and wireless ad-hoc network creation within RFID system, thus extending the potential applications of RFID. With all these consideration, this PhD work focuses on RFID as the identification, sensing, and communication technology for IoT.

1.3.1.1 RFID system components

Depending on the type of application as well as the industrial sectors that deploy it, an RFID system can be very complex. An overall RFID system components that can cover the very complex type of RFID application, consists of the following three subsystems [34]:

- *Radio Frequency (RF) subsystem*: This part performs identification and related transactions using wireless communication.
- *Enterprise subsystem*: It consists of specialized software that can store, process, and analyse data acquired from RF subsystem transactions to make the data useful to a supported business process.
- *Inter-enterprise subsystem*: This subsystem connects enterprise subsystems when information needs to be shared across organizational boundaries.

Essentially, the actually RF subsystem consists of readers, which are also called interrogators, and tags, which are also called transponders. Both the enterprise and inter-enterprise subsystem together are also commonly called as an RFID middleware, that consists of several components designed in a modular fashion. While the RF subsystem and the enterprise subsystem – to a certain extend, depending on the RFID middleware component used in the deployment – are the must have ones in every RFID system, an RFID system with an inter-enterprise subsystem is typically used, for instance in a supply chain application. Thanks to the inter-enterprise subsystem, a tagged product in a supply chain application can be tracked throughout its life cycle, from manufacture to final purchase, and sometimes even afterwards (e.g., to support targeted product recalls) [34]. Furthermore, with the tracking feature enabled by the inter-enterprise RFID subsystem, it can provide information and status of the tagged product or object, i.e. what, where, and why, prevent counterfeiting, enhance traceability, offer added value services, and so on. In the following sub-sections, we will be explaining the RF subsystem which consists of two main components, namely RFID tag and reader, based on the legacy standards and available commercial products. Thereafter, the RFID middleware, i.e. enterprise and inter-enterprise subsystems, will be described in more details in Section 1.3.2.

1.3.1.2 RFID Tag

An RFID tag must consists of two basic parts, namely the microchip and the antenna. The microchip in an RFID tag is a small electronic device that stores a unique identifier and may have also other features such as memory to store additional data and security algorithms. The antenna enables an RFID tag to communicate and exchange some data with RFID reader through the radio signal. In addition to those basic parts, an RFID tag may also comprise of a battery and sensor. The different components that made up a tag, may define different classifications of RFID tag.

In general, RFID tags can be classified based on the following features or criterion: power source, operating frequency, and identifier format. Table 1.1 summarizes features of RFID system operating at different frequencies based on ISO/IEC standard which also shows different power source modes, i.e. passive or active, and their potential usages and applications.

Table 1.1: Summary of RFID systems operating at different frequencies

< 135 kHz	**13.56 MHz**	**433 MHz**	**860-960 MHz**	**2.45 GHz**
Type A (FDX): operate at 125 kHz; permanently powered by the reader **Type B** (HDX): operate at 134.2kHz; powered by the reader only in the forward link	**Mode 1**: Reader-to-Tag data rate is 1.65 or 26.48 kbps; Tag-to-Reader data rate is 26.48 kbps **Mode 2** (high speed): Reader-to-Tag data rate is 423.75 kbps; Tag-to-Reader data rate is 105.9375 kbps on each of 8 channels **Mode 3** (high speed): **Option 1** (ASK based): Reader-to-Tag data rate is 26.7 - 100 kbps; Tag-to-Reader data rate is 424 or 848 kbps, and 53 - 212 kbps **Option 2** (PJM based): Reader-to-Tag data rate is 212 kbps; Tag-to-Reader data rate is 105.9375 kbps on each of 8 channels	Active RFID with reading range greater than 1 meter	**Type A**: Reader-to-tag data rate is 33 kbps; Tag-to-reader data rate is 40 or 160 kbps **Type B**: Reader-to-tag data rate is 10 or 40 kbps; Tag-to-reader data rate is 40 or 160 kbps **Type C**: Reader-to-tag data rate is 26.7 - 128 kbps; Tag-to-reader data rate is 40 - 640 kbps, or 5 - 320 kbps **Type D**: it is a tag talks only after listen technology	**Mode 1**: passive FHSS backscatter or narrow band operation **Mode 2**: data rate of R/W tag is up to 384 kbps and 76.8 kbps for R/O tag; active RFID with backscattering

1.3.1.3 Identifier format

Every tag has an identifier (ID) that is used to uniquely identify it. Since the RFID is used a global trading system which involves many companies and stakeholders, there needs to be a standard ID format, such that the encoding and decoding of the tags ID can be done across domains and organizations. The organizations that are actively involved in developing tag ID format in the context of EPC are the EPCglobal and GS1 (Global Standards One). The

results of their work on this matter are collected as part of the EPC Tag Data Standard (TDS) [29], which comprises a number of data formats for encoding and decoding tags ID. [29] also covers the specification of EPC and its relationship with GS1 keys and other existing codes. In general, the data structure of an EPC tag ID scheme consists of three or four fields as follows (see also Figure 1.1):

- The *Header* which specifies the EPC scheme or type
- The *General Manager Number* in GID, or the *GS1 Company Prefix* in the other EPC schemes. *General Manager Number* field is uniquely assigned by the EPCglobal and it identifies an organizational entity (essentially a company, manager or other organization) that is responsible for maintaining the numbers in subsequent fields, while *GS1 Company Prefix* is assigned by GS1 to a managing entity or its delegates.
- The *Object class* or *reference*. Depending on the usage of the EPC scheme, this field can be called by different terms, e.g. item, service, document, etc. This is essentially used by an EPC managing entity to identify a class or "type" of thing.
- The *Serial number* is a unique, non-repeating code within each of the object class. Please note that some of the EPC schemes do not have this field, e.g. GIAI and GSRN.

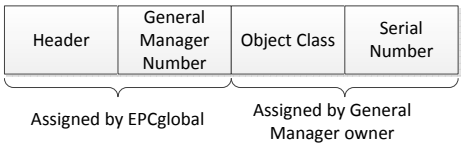

Figure 1.1: EPC tag ID format

It is important to note though that there are many cases where the RFID vendors are using their own tag ID data format in some commercial products; the organizations that choose to develop and implement their own tag ID format; and other networking protocols using different ID data format that are integrated to the RFID system. Therefore, various ID data formats in the tags produced by different vendors or other networking protocols, e.g. IP, need to be mapped to the standard EPC format, if te organizations that use those proprietary tag ID formats want to make their business to be compatible with the EPC network. To solve this issue, a translation process from those "unstandardized" ID formats to the EPC TDS need to be done at the middleware level. EPCglobal also provides a standard for this translation mechanism

known as GS1 EPCglobal Tag Data Translation (TDT) [30], which also serves the purpose of supporting the future development of EPC identifiers scheme. Issues related to the translation of the proprietary tag ID or other protocol ID format to the standard EPC format and how the results of the implemented solutions are, will be further discussed in Chapter 3.

1.3.2 IoT/M2M Middleware: Open RFID Middleware platform

The middleware is an important component in the IoT/M2M architecture, which essentially is a software layer mediated between the physical world and the application layers [9]. One of its features is hiding the details of different identification, sensing and communication technologies in the physical world, such as RFID, Zigbee, Bluetooth, etc, is fundamental to accommodate the IoT/M2M application developers from their lack of knowledge about certain communication technology, and accelerates the development process. Furthermore, it also has a major role in simplifying the integration of legacy technologies into new ones, especially for companies that want to adopt the IoT/M2M into their existing IT system.

The middleware that bridges the RFID and the rest of the application layers as well as the global RFID network is defined by by a set of standards released by EPCglobal, also known as EPCglobal standard. EPCglobal standard is an industry-driven standard that assists the development of EPC to support the spread use of RFID in the worldwide trading business, including supply-chain and logistic industries. It consists of interfaces specification as well as functionalities that an RFID middleware should have, and meant to be a guideline for any RFID implementation. The rest of this section will be dedicated to describe the EPCglobal architecture framework that covers all the standards on EPCglobal based RFID middleware and the existing RFID middleware implementation based on EPCglobal, along with its components and functionalities.

1.3.2.1 EPCglobal architecture framework

"The EPCglobal Architecture Framework is a collection of interrelated hardware, software, and data standards ("EPCglobal Standards"), together with shared network services that are operated by EPCglobal, its delegates, and others ("EPCglobal Network Services"), all in service of this common goal

[28]". EPCglobal itself is a leading organization in regards of industry-driven standards development for the EPC to support the use of RFID.

EPCglobal Architecture Framework is essentially an architectural guideline or standard that works as an open system to end users and technology vendors seeking to implement EPCglobal Standards and to use EPCglobal Core Services. Thus, it does not define a system architecture that end users must implement, nor does it dictate particular hardware or software components an end user must deploy. Figure 1.2 depicts the EPCglobal standards data flow relationships under EPCglobal architecture framework for Inter-Enterprise scenario which comprises different EPCglobal standards including parts of RFID middleware, hardware, and interfaces that inter-connect among different components within the same enterprise domain as well as inter-enterprise.

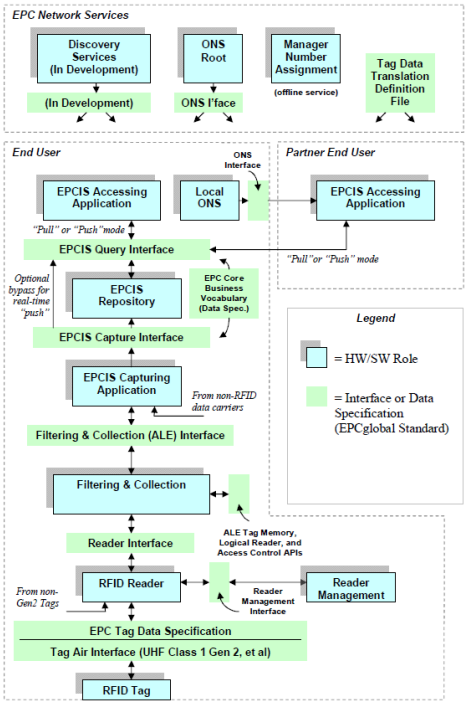

Figure 1.2: EPCglobal overall architecture, components and layers [12]

1.3.2.2 Existing RFID middleware

Essentially, an RFID middleware may be developed by any party or company in ICT industry either IT companies, RFID vendors, in house built in by the industries that apply RFID technology, or any open source initiative. As stated earlier that EPCglobal architecture framework does not define specific

system architecture, software, or hardware to be used in RFID middleware implementation, therefore RFID solution that is brought by each provider may varies depending on for instance the inter-operability with their existing IT solution.

One of the existing open source RFID middleware based on EPCglobal architecture framework is ASPIRE [22]. Figure 1.3 displays the ASPIRE project version of RFID middleware architecture based on EPCglobal architecture framework. ased on the depicted figure, ASPIRE adopts and implements Reader Protocol (RP), Low Level Reader Protocol (LLRP), Application Level Events (ALE), and Electronic Product Code Information System (EPCIS) module from EPCglobal standards.

Among others, the main parts that perform core functionality of RFID middleware are ALE server and EPCIS. Basic functionality and features of those two RFID middleware modules plus Object Name Server (ONS) will be discussed in further detail in the following sub-sections.

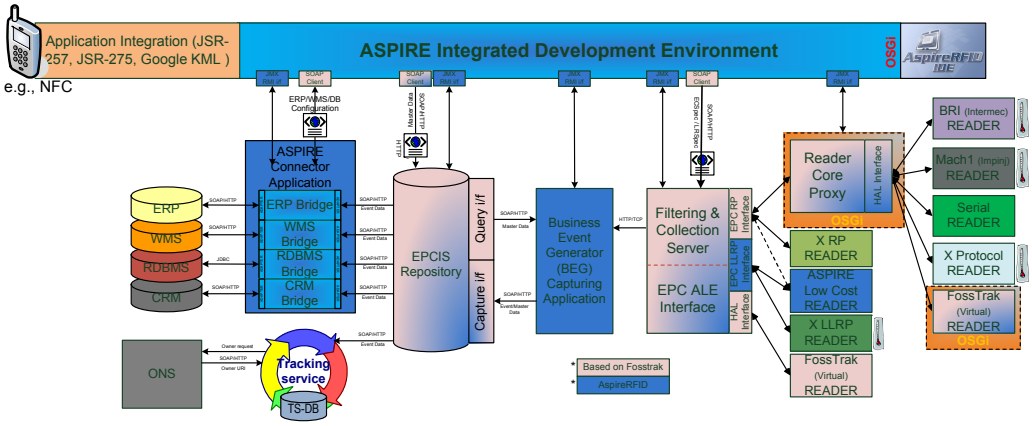

Figure 1.3: RFID Middleware architecture of ASPIRE based on EPCGlobal architecture framework [7]

1.3.2.3 ALE server

According to ASPIRE RFID middleware architecture (see Figure 1.3), the main parts of ALE server which define its functionality are as follow:

- **Hardware Abstraction Layer (HAL):** ensures inter-vendor interworking since many RFID readers come with proprietary standards

- **Filtering and Collection (FC) layer:** allows collection of specific RFID tag ID pattern (after this point, *RFID* tag will be referred as *tag* only), and filter out any unneeded ID due to multiple read of the same tag ID or simply because of the application's need.

HAL is an interface that allows RFID reader, and even any type of reader or device, for instance bar-code reader, NFC, bluetooth, or WSN, to communicate with middleware regardless communication protocol that they are using, i.e. RP and LLRP which are based on EPCGlobal standard, or any proprietary protocols. In the real implementation, in order to integrate a particular RFID reader or other device which uses communication protocol other than that defined by EPCGlobal standard, a HAL for that device need to be developed and be part of ALE server. It is worth to mention that HAL is one of extension component from the original EPCGlobal standard in ASPIRE RFID middleware.

As mentioned earlier, FC layer is a layer that performs collection particular RFID tag ID pattern and filter out any unnecessary tag ID which can be read by multiple readers or antennas in the vicinity of the tag and reduce the data traffic of unnecessary information in the higher layer. Additionally, filtering in RFID middleware is necessary simply because the application requires it. For instance, an RFID reader is placed in the dock door of a warehouse to monitor how many items have been received. In that case, the application needs to read a specific ID only once, i.e. the first read, to understand that the item has been received even though the reader might read multiple times.

1.3.2.4 Business Event Generator (BEG)

It automates the mapping between reports, i.e. ECReport (Event Cycle Report), produced by FC layer and EPCIS events [58]. In EPC terms, BEG can be seen as a specific instance of EPCIS capturing application that parses ECReports, fuses these reports with business context data.

1.3.2.5 EPCIS

There are 3 main parts of EPCIS that are defined in the EPCGlobal specification namely EPCIS capture interface, EPCIS repository, and EPCIS query interface.

EPCIS capture interface is defined as an interface that responsible to provide a path for communicating EPCIS events generated by EPCIS Capturing

Applications to other roles that require them, including EPCIS Repositories, internal EPCIS Accessing Applications, and Partner EPCIS Accessing Applications. In the actual RFID Middleware, EPCIS capturing application that interprets the captured RFID data from the lower layer of Middleware, e.g. FC layer or other component depending on the implementation, needs to be implemented along with the EPCIS capture interface.

EPCIS repository is a database system that provides persistence information not only raw RFID data, but also other RFID related events which are defined in FC layer as well as business context information. Those data can then be accessed by accessing application, both internal and external, through EPCIS query interface as defined in EPCGlobal specification.

In principle, the main functionality of EPCIS component in an RFID middleware based on EPCIS standard [26] is as follow:

- Receive application-agnostic RFID data from the ALE server through the BEG component (see Figure 1.3).
- Translate RFID data in corresponding business events. These events carry the business context as well (i.e., they refer to particular companies, business locations, business processes etc.).
- Make business events available and accessible to other upstream applications.

In general, the ASPIRE EPCIS repository is built to deal with two kinds of data, namely:

1. RFID event data i.e. data arising in the course of carrying out business processes. These data change very frequently, at the time scales where business processes are carried out.
2. Master/company data, i.e. additional data that provide the necessary context for interpreting the event data. These are data associated with the company, its business locations, its read points, as well as with the business steps comprising the business processes that this company carries out.

It is worth to mention that in addition to two above mentioned points, the ASPIRE EPCIS repository also deals explicitly with historical data (in addition to current data) and operates within enterprise IT environments at a level that is much more diverse and multi-faceted comparing to the underlying

data capture and Filtering and Collection middleware components. Moreover, as stated earlier that EPCIS repository makes the business events available and accessible to other upstream applications, i.e. applications sitting at the other enterprises, etc, the architecture in Figure 1.3 could be extended to include external enterprises, and security as well as privacy issues related to this scenario is even more exposed. The security and privacy issues on EPCIS repository following the Inter-Enterprise will be elaborated in chapter 2.

1.3.2.6 Connector

Connector is responsible to interface the EPCIS repository with the other legacy systems, e.g. corporate IT systems (CRM, ERP, database, etc). In telecommunication world, it is similar to the role of gateway.

1.3.2.7 ONS

The ONS is a service that returns a list of network accessible service endpoints that pertain to a requested EPC. The ONS does not contain actual data about the EPC; it only contains the network address of services that contain the actual data [58]. The ONS uses the Internet's existing Domain Name System (DNS) for resolving requests about an EPC. In order to use DNS to find information about an item, the item's EPC must be converted into a format that DNS can understand, which is the typical, "dot" delimited, left to right form of all domain-names. The ONS resolution process requires that the EPC being asked about is in its pure identity Uniform Resource Identifier (URI) form as defined by the EPCglobal TDS [27]. This URI conversion is done in the local server by the TDT component.

The key components of the logical architecture are the Core ONS servers. The core of the system comprises by the ONS servers themselves. Each company assigned part of the global EPC namespace is responsible to create and maintain a functional ONS server to serve queries about EPC's inside that namespace. Each ONS server is basically a standard DNS server with special Naming Authority PoinTeR (NAPTR) records [27], mapping EPC's to service access points. Typically EPCIS query and capture interfaces where additional information can be obtained about the EPC in question [58].

1.3.3 IoT/M2M Networking Architecture, Components, and Protocols

In this section, the networking aspect of IoT/M2M based on some proposed or existing works and initiatives that have been done to achieve the vision of IoT will be presented. As already discussed earlier that the network oriented vision of IoT is an important factor towards a successful IoT/M2M deployment. This vision leads to the necessity of developing and designing a good and efficient networking concept, comprising the architecture, components as well as protocols, for the IoT/M2M.

The authors of [60] proposed two types of network architectures for M2M called hierarchical network architecture for scalable connectivity and hierarchical networks for high capacity. Both of them are primarily concern about the networking infrastructures – especially wireless networks – i.e. WPAN/WLAN/Wireless Wide Area Network (WWAN), to support high demand and traffic load generated to and from M2M devices. The central idea of the first architecture is to increase the network scalability by preventing direct traffic from M2M devices to the Wide Area Network (WAN) through an M2M gateway that manages the M2M devices and serves as an aggregation point. Moreover, it also considers a peer-to-peer connectivity support depending on latency requirements and the type of information exchanged. On the other hand, the second architecture proposed multi-tiers and multi-radios architecture to increase the capacity of the network. Multi-tiers hierarchy consists of some macro base stations, e.g. in 3G/4G network, supporting large coverage and high mobility for M2M devices, coexisting with pico/femto base stations or WLAN access points that serves to improve link reliability and to increase system capacity. In order to take the full advantage of this multi-tiers hierarchy architecture, employing devices with multi-radio system is of a great favour. However, they do not touch the networking protocols aspect within the IoT/M2M devices themselves, as well as self-configuring and managing features that are urgently needed in IoT/M2M.

On the other hand, another great effort towards realizing a networking architecture and protocols platform in IoT/M2M has been carried out under the development of 6LoWPAN standard, through Internet Engineering Task Force (IETF) 6LoWPAN working group. The main reason why the 6LoWPAN is highly pushed as the networking protocol for IoT is due to wide acceptance of IP in the legacy networking devices and the fact that it supports the IP implementation on low-powered wireless devices, especially those that are using IEEE 802.15.4 standard and link layer or access technologies, e.g. sub-GHz ISM band radio and low-rate power line communication (PLC). Furthermore, it

also extends the already established Mobile Ad-hoc NETwork (MANET) into the multi-hop wireless mesh routing where an internet gateway is introduced in the topology. Besides, those aspects on link layer protocol adaptation for low-power devices and routing protocol, 6LoWPAN defines some standards on network management, neighbour discovery, and mobility, i.e. adapted from Mobile IPv6. The 6LoWPAN architecture is also defined, which is made up of several Low-power Wireless Personal Area Network (LoWPAN) and consists of Simple LoWPAN, Extended LoWPAN, and Ad hoc LoWPAN [55]. Figure 1.4 shows three types of 6LoWPAN architectures.

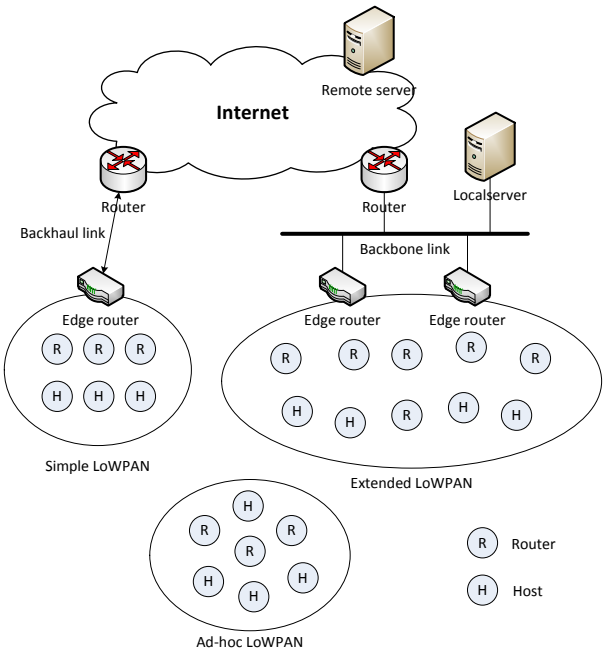

Figure 1.4: The 6LoWPAN architecture

Another initiatives on attaining the networking architecture and protocols in IoT/M2M has been performed in the context of PN within MAGNET project [47]. Similar to the 6LoWPAN, [47] has done a huge contribution on defining networking protocols for device-to-device, i.e. IoT/M2M, including the infrastructure based and ad hoc architectures. However, the [47] focuses a lot on the development of PN, and further the PN-F concepts, extending the Personal- Personal Area Network (PAN) and/or WPAN, along with self-organizing mechanism, service management, and context management. Furthermore, security mechanisms as the key in creating secure PN and PN-F, through long term trust establishment by means of public key and PN certificate generation, have also been thoroughly researched and defined, particularly in the context of PN and generally for the device-to-device communication. The overall PN architecture consists of three abstraction levels view, namely

connectivity abstraction, network abstraction, and service abstraction levels. Within the network abstraction level, three types of communication architectures are defined: intra- and inter-cluster communication (within the same PN), and federated communication, i.e. among different PNs. Figure 1.5 depicts an example of the PN architectures.

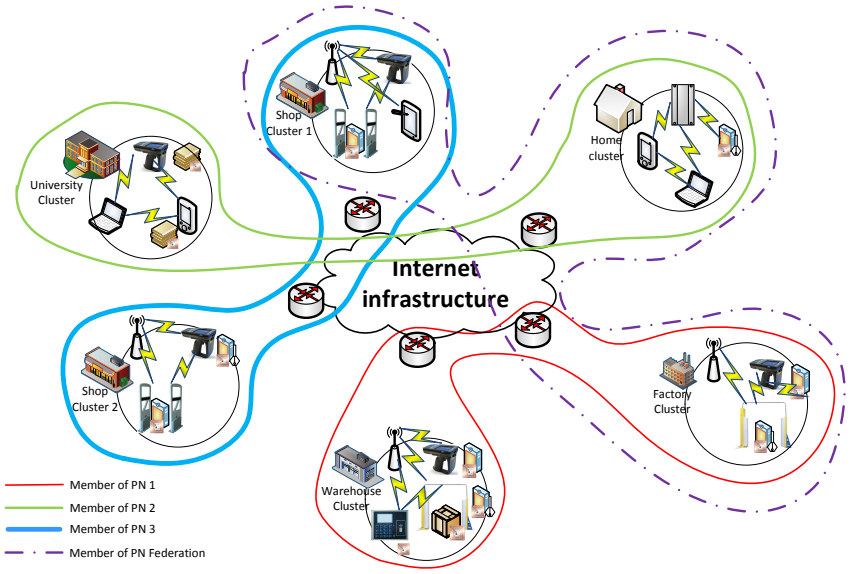

Figure 1.5: Example of PN architecture

From Figure 1.5 we can see that the intra-cluster communication is done within the home, office, or hotel cluster, e.g. using WPAN interfaces of wireless personal devices within the proximity. Furthermore, the inter-cluster communication can also be seen in the communication between hotel cluster and home cluster in PN 1, or between airport cluster and office cluster in PN 2. In this type of communication, a logical entity called PN gateway in each of the cluster is used. The PN gateway can physically be implemented in an edge router, which is similar to the 6LoWPAN approach, or other routing capable devices. Finally, PN Federated communication can be seen in Figure 1.5, among the clusters and PNs within the orange line. In addition to the types of communication mentioned previously which rely on internet infrastructure, the ad-hoc based PN can also be established as can be seen among the User 4, 5, and 6 in Figure 1.5.

1.3.4 IoT/M2M Cloud Platforms

1.3.4.1 Cloud Computing Paradigm

Cloud computing has rapidly grown a huge attention in recent years due to its great flexibility, scalability, and availability in obtaining computing resources at lower cost [32]. As cloud computing is an emerging form of distributed computing that is still undergoing evolution and standardization, the term itself is often used with different meanings and interpretations [32]. Nevertheless, cloud computing system has some important characteristics and paradigms which lead to a general taxonomy that describes different aspects of it.

1.3.4.1.1 Users and providers relationship The users and providers relationship in cloud computing is more complicated than that in general web services, consisting of three roles, i.e. Cloud System provider, Cloud Service Provider, and Service Consumer, as depicted in Figure 1.6 [64]. The *Cloud Service Consumer* is an individual or organization that consume the Cloud Computing service provided by *Cloud Service Provider*. The *Cloud Service Provider* is the organization that offers the Cloud Computing service, whom is also a user of Cloud Computing system. The *Cloud System Provider* is the organization that offers the Cloud Computing system or infrastructure, which may be a corporate business, non-profit organization, or government agency. In this type of relationship, it is important to note that *Cloud Service Provider* acts as third party that maintains information about, or on behalf of, another entity [64].

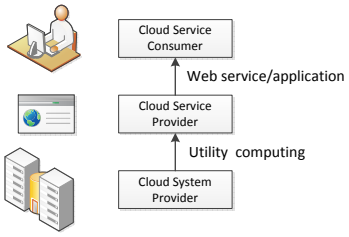

Figure 1.6: Users and providers relationships in cloud computing

1.3.4.1.2 Essential characteristics Five essential characteristics of cloud services that demonstrate their relation to, and differences from, traditional computing approaches are as follows [17]:

- **On-demand self-service.** A consumer can unilaterally provision computing capabilities such as server time and network storage as needed automatically, without requiring human interaction with a service provider.
- **Broad network access.** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs) as well as other traditional or cloud based software services.
- **Resource pooling.** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a degree of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources, but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines. Even private clouds tend to pool resources between different parts of the same organization.
- **Rapid elasticity.** Capabilities can be rapidly and elastically provisioned — in some cases automatically — to quickly scale out; and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- **Measured service.** Cloud systems automatically control and optimize resource usage by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, or active user accounts). Resource usage can be monitored, controlled, and reported — providing transparency for both the provider and consumer of the service.

1.3.4.1.3 Cloud deployment models

- **Private cloud**: Data and processes are managed within the organization without the restrictions of network bandwidth, security exposures and legal requirements that using public cloud services across open, public networks might entail.

- **Public cloud**: It describes the cloud computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, self-service basis over the internet, via web applications/web services, from an off-site third party provider who shares resources.
- **Community cloud**: The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, or compliance considerations). It may be managed by the organizations or a third party and may exist on-premises or off-premises.
- **Hybrid cloud**: The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

While the choice of deployment model has implications for the security and privacy of a system, the deployment model itself does not dictate the level of security and privacy of specific cloud offerings. That level depends mainly on assurances and the extent of visibility into performance and management details of the cloud environment, which are furnished by the cloud provider or independently acquired by the organization [32].

1.3.4.1.4 Cloud service models Cloud service models are another important consideration in cloud computing just as cloud deployment models do. The service model to which a cloud conforms dictates an organization's scope and control over the computational environment, and characterizes a level of abstraction for its use. A service model can be actualized as any of deployment models [32]. The three fundamental classifications of cloud service models are as follows (see also Figure 1.7):

- *Software as a Service (SaaS)*: SaaS model allows the cloud service consumer to use the cloud service provider's applications running on a cloud infrastructure through various client applications and devices. The consumer does not manage or control the underlying cloud infrastructure, i.e. network, servers, OS, storage, or individual application capabilities, with the possible exception of limited user specific application configuration settings [17]. It is a multi-tenant platform that uses common resources and a single instance of the object code of an application as well as the underlying database to support multiple customers simultaneously [50].

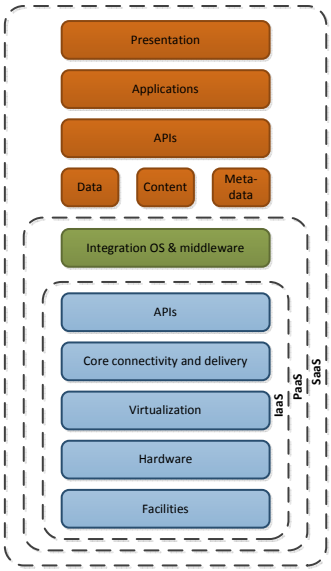

Figure 1.7: Cloud service models

- *Platform as a Service (PaaS)*: PaaS model allows the consumer to deploy the consumer-created or acquired applications created using programming languages and tools supported by the cloud service provider onto the cloud infrastructure. A platform including all the systems and environment comprising the end-to-end life cycle of developing, testing, deploying, and hosting of sophisticated web applications as a service, is provided to the developers (or consumer) [50]. With hundreds of tools and services readily available in the platform, the development time can be greatly reduced and quickly scaled. The consumer has only control over the deployed applications and possibly application hosting environment configurations, but has no ability to manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage [17].
- *Infrastructure as a Service (IaaS)*: IaaS model allows the consumer to provision processing, storage, networks, and other fundamental computing resources where the consumer can deploy and run arbitrary software, including operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls) [17]. Besides the higher flexibility, a key benefit of IaaS is the usage-based payment scheme which allows consumers to pay as they grow. Another

important advantage is that by always using the latest technology, customers can achieve a much faster service delivery and time to market [50].

1.3.4.2 Existing IoT/M2M Cloud Platforms

There are numbers of IoT/M2M Cloud Platforms offered by companies as their commercial product or service. Some examples are Axeda M2M Cloud Service [10], Xively [63], Libelium [38], etc. Research projects that aims particularly at developing a IoT/M2M Cloud platform and solving some research problems at the same time, e.g. reliability, scalability, security, etc, are also present. One example of such research project is Building the Environment for Thing as a Service (BETaaS) project [12]. In what follows, some overview of the existing IoT/M2M Cloud platforms from Xively will be briefly presented in this subsection.

Xively is a PaaS that provides the tools and services needed by developers to connect sensor data (e.g. energy and environment monitoring of a building, object, etc) to a public cloud and to create compelling applications based on that data. Figure 1.8 shows a basic overview of Xively Cloud Services [63].

As illustrated in Figure 1.8, Xively Cloud Services provide messaging, data archiving, device provisioning, and directory services which are accessible through the Xively Application Programming Interface (API). Xively's web applications leverage the API to provide connected product lifecycle management capabilities through Xively Developer Workbench and Xively Management Console [63]. At the bottom layer, the Xively API provides interfaces to connected objects (e.g. sensors and hardware platforms), applications (e.g. web and native), and customer backend services (e.g. Customer Relationship Management (CRM)), in accessing Xively Cloud Services that is based on RESTful services, through different protocols, such as HTTP/HTTPS, socket/websocket, and Message Queuing Telemetry Transport (MQTT). On top of that, various libraries are provided as Open Source to allow painless integration between different programming languages or device platforms and the Xively API. Currently, the supported libraries are as follows: Android, C, Objective-C, Java, Java Script, PHP, Python, Ruby, Arduino, ARM mbed, and Electric Imp.

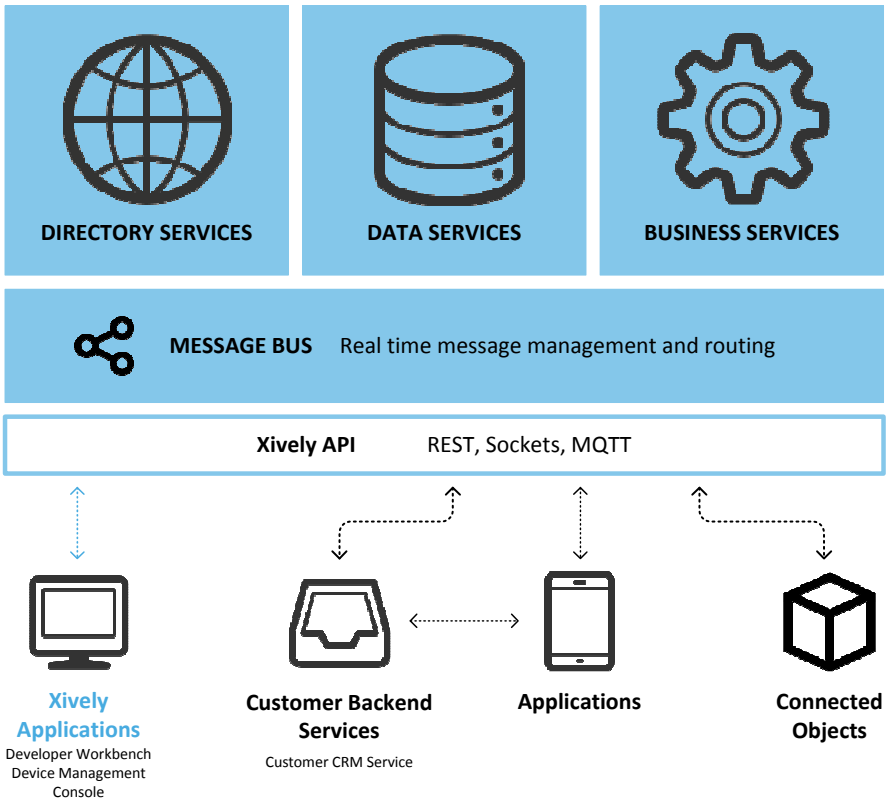

Figure 1.8: Overview of Xively Cloud Services

1.3.4.3 Security models in IoT/M2M Cloud Platform

In order to draw a security model of a IoT/M2M Cloud Platform, it is important to review some of the existing IoT/M2M architectural reference models, such as IoT-A [31], ETSI M2M [18], Constrained Application Protocol (CoAP) [56], and the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) Focus Group (FG) M2M [40]. But as far as the security model is concerned, the reference model of the IoT-A provides the most concise and comprehensive security model. Hence it is going to be the focus of the existing IoT/M2M security model review in this section. In addition, the security mechanisms used in Xively will also be briefly presented in order to give a comparison with the general security model introduced by IoT-A.

The security model in IoT-A consists of three main concepts: Trust, Privacy and Security [31]. Those functional blocks are horizontal and also tightly related among themselves. Figure 1.9 security features and general layering of security features in IoT-A [31]. Trust in an IoT environment can be split

into two layers: networking and application. Networking trust is related to the integrity of the routing process while application trust is more related to information quality, such as endpoint authentication and non-repudiation; confidentiality; privacy policy. Privacy is associated with the capability of a person to protect their information, moreover by a legal perspective laws can put restrictions and obligations on the way information concerning users is stored and used. Security is a general concept that incorporates also Trust and Privacy [31].

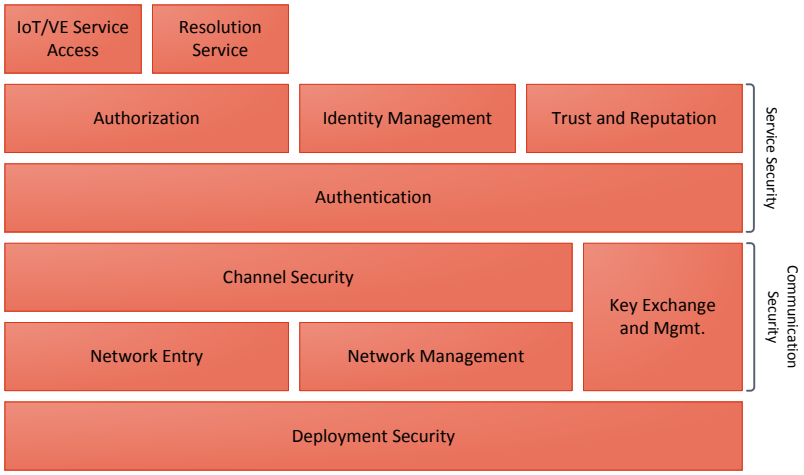

Figure 1.9: IoT-A security features

To accomplish those concepts, the information model should include descriptions of access policies, certificates and trusted identities. Moreover the functional architecture should include components responsible for the management of trust, security and privacy.

The definition of the model for communication security adopts the distinction between constrained and unconstrained networks from the communication model. In fact, the heterogenity of devices represents a problem in designing a security model. To overcome this issue the security model has a high level of abstraction. Gateways, being at the edge between networks of the different network types, are identified as central components for mediating between different security components of different networks. This includes tunnelling, filtering, and, most importantly, scaling down security (and other) functionalities from unconstrained networks in order to meet the limitations imposed by the constrained networks [44].

In Xively, the platform and web tools run on LogMeIn's proprietary secure cloud infrastructure [39], to ensure the service and client's data is protected,

available, and fault-tolerant. Several security aspects emphasized in Xively, especially for the developers, will be briefly described as follows [62]:

1. **API keys:** The Xively API uses API Keys to authenticate requests and to control access to all API resources. An API key is a hierarchy of three objects: key object, permissions object, and resource object. A key object is the main object of an API key. This key object must consist of at least a permissions object which represents a different set of permissions, while it is optional to have one or more resource object within a permissions object. A fine-grained access control can be realized by the API keys by configuring value of the permission object according to the developer's needs. Thus, the access to a resource can be granted on an as-needed, when-needed basis, and take it away when it's not.

2. **OAuth:** OAuth [46] is a method for allowing third-party application access to a resource owner's resources without giving them access to resource owner's username and password. The Xively implementation of OAuth 2.0 currently gives the third party site an API key which can access resource owner whole account with configurable permissions. Using OAuth in Xively is excellent for tying in administrative applications and services, such as an application that is used by a company to monitor all of its customers electricity usage and integrate it with an existing management system, e.g. CRM. It is also good for an application in which the users interact in a collaborative manner. However, it is not recommended for applications in which the end users can interact with their device because OAuth will give permission or authority to users to access any feed or information under the account's owner.

3. **Secure provisioning:** A device is provisioned with a Feed ID and API Key upon its first Activation call to the Xively API. The flexible activation processing infrastructure with secure and encrypted device provisioning allows the user to automate what happens when the user's devices wake up for the first time. This allows calibration of secure connection is established, credentials are exchanged, and devices, applications and services into user's circle of trust.

1.4 Challenges of the IoT/M2M System

1.4.1 Security and Privacy

1.4.1.1 The number of vulnerable systems

The number of vulnerable systems and attacks vectors will surely increase in the context of the IoT, thus fault tolerance becomes essential. Not only we must strive for security by default (robust implementations, usable systems, etc.) in the IoT, but also we need to develop awareness mechanisms that can be used to create the foundations of intrusion detection and prevention mechanisms, which will help IoT entities to protect or even gracefully degrade their services [51].

1.4.1.2 Access Control

Rolf H. Weber brings up the new security and privacy challenges in [59] in which he points out that the technical architecture of IoT has an impact on the security and privacy of the involved stakeholder. Among others, one of the security and privacy requirements mentioned in [59] is the *Access Control*, which necessitates that the information providers must be able to implement access control on the data provided. Further on, [59] also mentions how the access control has to be implemented in the presence of Peer-to-Peer (P2P) system or Distributed Hash Table (DHT) which can be employed to increase the security and privacy.

The authors of [9] also discuss the security and privacy challenges of the IoT quite extensively. Apart from the types of security threats imposed to the IoT, such as physical and eavesdropping, as well as the major security concern on the authentication and data integrity whose solutions are mostly based on cryptographic methods, privacy is another major issue that can be looked and solved based on different approaches. One of the approach in dealing with the privacy issue is to ensure that the personal data collected by the IoT system is used only to support the authorized services by authorized providers [9], which essentially suggests the proper usage of access control system.

From the cloud computing side, NIST published a set of guidelines to ensure the security and privacy especially for the public cloud deployment in [32]. In summary, the key security and privacy issues in cloud computing are as follows: governance, compliance, trust, architecture, identity and access management, software isolation, data protection, availability, and incident response. Again,

access control – which comes under identity and access management – is listed among the key issues to be taken care of also in the cloud computing. It is important to note that, the security and privacy risks in the cloud computing services vary significantly among organizations, depending on the intended purpose, assets held, legal obligations, and so on [32]. Therefore, the security and privacy objectives of each organization is the key factor in determining the appropriate mechanisms for securing their cloud services. In the context of access control, it is necessary to have a flexible access policy framework that can cope with this challenge, i.e. compatible with various security and privacy objectives of different organizations.

Last but not least, the authors of [51], and [20] in particular, highlight the importance of resource and information protection, i.e. access control, in distributed systems such as the IoT. With this, we can safely conclude that the *Access Control* is one of the important security and privacy concerns in IoT/M2M while its actual model and implementation in the IoT/M2M systems is often overlooked and remains as an open issue. Unique characteristics of the IoT/M2M system and architecture signify particular challenges in the access control of such system. These issues will be further discussed in Section 1.4.2.

1.4.2 Distributed architecture

As have been discussed earlier, the vision of IoT/M2M to connect billions and even trillions of devices over the internet will eventually make the communication topology or architecture be more and more distributed. Although the early adopted approach of IoT/M2M is the centralized approach, the distributed architecture approach of IoT/M2M is a necessity in order to take full advantage of it. Rodrigo Roman et al. in [51] analyse the features and some specific security and privacy challenges of distributed IoT. In summary, [51] specify some challenges and considerations specific to access control of the distributed IoT, such as scalability and consistency, granularity, constrained devices, context, and delegation mechanisms.

1.4.2.1 Scalability

The scalability and consistency issues will arise when storing the list of users and their associated access rights in Access Control List (ACL)s [51] due to the huge numbers of nodes and users interacting to each other in the IoT/M2M system. Employing an existing access control mechanism, e.g. the Role Based Access Control (RBAC), is not straight forward as it needs to define roles that

users can take, which might be different in various contexts even though they refer to the same type of entity [51]. Indeed, the Identity-based access control, such as RBAC, have been found to be inflexible, don't scale well, and are difficult and to upgrade [33]. Hence, designing an access control mechanism that can cope with the scalability challenge a distributed environment such as IoT/M2M is of upmost important, which is not as trivial as applying the existing access control mechanisms.

1.4.2.2 Context

IoT is closely connected to its context [11]. The context or contextual information includes environment context (e.g. sensing data), temporal context (e.g. time and location), user context, system context, and some specific business context (e.g. information about business step, business transaction type, etc, in RFID middleware). However the context is inherently dynamic in nature which will influence the access policy decision [11]. It is also reported in [21] that one of the requirements of the access control solution for IoT is to be flexible to adapt to different contexts. In fact, the IoT/M2M is the dynamic nodes connectivity and network topologies due to the mobility and other factors such as power constraint. Therefore designing a flexible access control system that can cope with the dynamic nature of IoT/M2M and contextual information is one of the challenging tasks.

1.4.2.3 Delegation

Another characteristic of a distributed or decentralized system is the decision making can be delegated from the higher authorities to the lower level entities. In a totally distributed topology, i.e. without hierarchy, the authority delegation can be performed among the entities. As earlier mentioned, the dynamic nature of IoT/M2M is an inherent challenge in such a system, hence the authority delegation is another way of coping with this characteristic. For example, a node may delegate its authority to the other nodes before it dies out, e.g. out of battery. The challenge for access control in IoT/M2M in this regard is how to design a dynamic and efficient authority delegation which is able to keep the delegation process under control, e.g. not delegating access just to any entity.

1.4.2.4 Granularity

The authors of [51] also indicate the importance of granularity (i.e. providing more information to people with the right credentials) and location (i.e. whether users are accessing the thing services locally or remotely) for the access control policies in certain scenarios. But actually it should be generalized that not only the location, but the context along with granularity are two important elements for the access control policies in IoT/M2M, as the context that are attached to the users and resources being accessed will determine the granularity level of the information to be disclosed to the users. The real challenge in this aspect is how to design a fined-grained access control policy with the help of contextual information to deal with large amounts of information, including the measurements data and events, generated by IoT/M2M nodes.

1.5 Problem statement and research questions

As the attempt to answer the challenges explained in Section 1.4, the research problems addressed in this PhD dissertation is on the design of a scalable, flexible, and fine-grained access control mechanism for the IoT/M2M Cloud platform by making use of distributed approaches for resource constraint networks such as RFID and wireless sensor network. The problem statement is divided into different sub-problems as follows:

- Management and security access control in IoT/M2M middleware
- Access control and authority delegation in federated IoT environment
- Intrussion detection system in the access control for the IoT/M2M Cloud platform

The previously defined problem statement leads to the following research questions:

- How to provide a fine-grained, efficient policy assignment, and interoperable access control in IoT/M2M middleware, particularly the RFID middleware?
- How to support the efficient and scalable mobility management coupled with the access control in the IoT/M2M middleware?

- How to overcome the scalability and dynamic nature of the IoT/M2M issues, and to provide a certain level of authentication in the access control model?
- How to model the intrusion detection system incorporated with access control system for the IoT/M2M Cloud platform, where there exist a set of security assets to be protected, the uncertainty on the maliciousness of the opponent, and some resource constraint?

1.6 Hypotheses and research methodology

1.6.1 Hypotheses

The research questions regarding the access control in the IoT/M2M Cloud platform as described previously in Section 1.5 lead to several hypotheses described as follows:

1.6.1.1 Fine-grained access control

An access policy design that incorporates access rule evaluation that consists of specific RFID events attributes and conditions as well as the user profiles and trust can fulfil the requirements of fine-grained and efficient access policy assignment in the access control model for the inter-enterprise RFID system, i.e. RFID middleware. In addition, a web service based implementation of the access control system allows interoperability with the overall RFID middleware.

1.6.1.2 Efficient mobility management and access control

An architectural approach is needed in order to integrate the RFID system to the global IP network and providing an efficient mobility management at the same time. Moreover, it can also de-couple the access control system in order to make it more scalable. For such reasons, a hypotheses is made that a SIP-based location management system in the RFID middleware can provide seamless integration between RFID system to the global IP network, and efficient mobility management and access control.

1.6.1.3 Capability-based and context aware access control

The scalability issue in the IoT/M2M networks which consists of huge numbers of devices or "things" is addressed by using a capability-based access control as it relies on a capability (also known as a "ticket") owns by each entity in a distributed manner. Including some context information and a delegation framework in the access control model will help the system to make a more precise access decision and provide flexibility against the dynamic nature of the IoT/M2M, e.g. dynamic context and connectivity due to mobility and power constraint. Based upon these arguments, the following hypotheses is made: the capability-based and context aware access control with a delegation framework will be able to overcome the scalability issue and the dynamic nature of the IoT/M2M. As a capability comprises a token that needs to be validated prior to granting an access, another hypotheses is made that the capability-based and context aware access control can also provide some level of authentication.

1.6.1.4 Intrussion detection in the access control system

An intrusion detection system is needed to be part of the access control system to detect and mitigate the threats, especially those that come from the insiders of an M2M local cloud platform. The interaction between an attacker and a defender equipped with intrusion detection can be modelled by the game theory. It is further argued that the uncertainties that exists about the maliciousness of the opponent can be modelled with a multi-stage Bayesian game which include an analytical model of a set of assets with different values to be protected by the defender and the resource constraint of each player.

1.6.2 Methodology

The high level overview of the research approach in this PhD work starts with identifying the root cause of the problems, followed by creating a high level model that bridges the problems and solution, then the proposed solutions are designed (e.g. complete design of protocol, software architecture, and mathematical model), verified (e.g. by means of computer simulation, software development and experiments), and analysed. The method used in this research is mainly the quantitative research method.

To carry out the quantitative research method, the data is collected through some computer simulations and experiments. The simulation tools used to carry out the computer simulation include MATLAB and AVISPA (a validation

tool of internet security protocol and application) that are widely accepted by the scientific community, and the detailed simulation procedure is explained later in the respective chapter. The experiment activities in the context of this PhD research work is carried out to follow up the software implementation of the proposed solutions, which results in a collection of measured data for further analysis. As a result, the qualitative analysis is performed based on the simulated and measured collected data, which is explained in more detail later in each of the respective chapter.

1.7 Research goals and scope

The background and motivation of the IoT/M2M and cloud computing systems as well as the security and privacy challenges therein that have been outlined so far in this dissertation lead to many interesting research questions and subjects of study. However, given the limitation of time and resources that constitute a PhD study, the scope of this dissertation is confined to investigating a number of security and privacy issues for a specific IoT/M2M technology and cloud computing model. In this section, we present the research goals as well as the scope and limitations of the work, and the rationale behind the delimiting choices.

1.7.1 Research goals

Based on the challenges and hypothesis stated earlier in Section 1.4 and 1.6, the goals of this PhD research are defined as follows:

1. Design and implementation of fine-grained access control in the scope of RFID middleware.
2. Design and implementation of a mobility management in RFID middleware that supports the access control mechanism.
3. Desgin and implementation of scalable access control with a flexible access delegation for the IoT/M2M Cloud platform.
4. Analysis and design of an intrusion detection in the access control system for the IoT/M2M Cloud platform.
5. Test bed implementation for proof of concepts towards real system realization of the research works.

1.7.2 Scope of the research

Most of the security and privacy issues that are researched and presented in this dissertation are related to the access control. This PhD study is part of several European Commission funded projects, such as My Adaptive Global NETwork (MAGNET), Advance Sensors and Programmable middleware for Innovative Rfid Enterprise applications (ASPIRE), Intelligent System for Independent living and SElfcare of seniors with cognitive problems or Mild Dementia (ISISEMD), and BETaaS, where security and privacy requirements are mostly focussed around the access policies among different types of users into the platform, middleware, or devices. With this background in mind, the focus of this PhD work was limited to access control mechanism among the stakeholders of the respective projects into the IoT/M2M and Cloud platform, regardless of any method applied to authenticate the user or entity. Moreover, the IoT/M2M is highly distributed and dynamic in nature. Such requirements have motivated us to focus the large part of our research on designing a scalable access control model with access delegation in multi-domain or federated environment; and also designing of a scalable IoT/M2M Cloud platform, including the middleware, in which such access control model are to be applied.

The target platform in terms of the identification, sensing, and communication technology in this PhD study is limited to the RFID and sensor technology. The maturity and advance development of RFID technology towards realizing the vision of IoT – besides the fact that part of this PhD study is done under the scope ASPIRE project where RFID becomes the central point of interest – have become our strong motivation to limit our focus in this regard. Having this decision in mind, the investigation of the middleware for IoT/M2M system in the scope of this PhD study is limited to RFID middleware which has been briefly introduced in Section 1.3.2.

The type of services offered by most of the existing IoT/M2M cloud service providers are generally either a *Software as a Service* (SaaS) or a *Platform as a Service* (PaaS) models. In this regard, we include a new type of cloud service model known as *Things as a Service* (TaaS), which contains a unified representation of an M2M model and enables any high level IoT/M2M service to be built. This is a concept that is introduced in BETaaS project, which also introduces another concept of cloud model where a set of distributed IoT/M2M gateways are able to form cloud locally. As pointed out earlier, a scalable access control that supports flexible access delegation is also a necessity in such a distributed local cloud scenario. Furthermore, an intrusion detection system needs to be part of the access control system to enable the monitoring of important resources, and to detect of any malicious behaviour. In this

study, the malicious behaviour monitoring is focused to that which is coming from within the local cloud, i.e. the insider, because it is a great security threat to the IT systems and network in general. Therefore, the study of dynamic interaction between malicious and regular gateway which has a set of accessible resources or security assets, under some constraint in terms of attack and monitor resources, is presented in this dissertation.

An idea remains an abstract concept unless it is implemented in a real system realization. Test beds and practical implementation of such an IoT/M2M Cloud platform along with the security access control mechanism is a necessary step in a life cycle of any novel research, i.e. to proof the proposed concepts and to take the ideas further into the real product. E-Health scenario is our first focus in the test bed implementation within the scope of this PhD work. The huge potential of eHealth to answer the challenge of ageing population and to reduce the health treatment cost, yet lacking in maturity and real products and services taking off in the market, have become our motivation to focus our test beds implementation in such area. The supply chain use case is another test bed implementation in this PhD work, which is motivated by the fact that the supply chain scenario is the main driver in the development of RFID middleware.

1.8 Overview of the dissertation and scientific contributions

An overview of the chapters and the scientific contributions that constitute this PhD dissertation are summarily discussed below.

In chapter 2, a proposed access control model designed to deal with a very large and highly dynamic inter-enterprise RFID data is presented, along with system implementation and evaluation. The key contributions of the proposed access control model are as follows. First, a dynamic and efficient access policy mechanism of an object or group of objects, based on the EPCglobal compliant attributes and vocabularies is introduced. This access policy takes the profile of the accessing entity (i.e. individual or organization that requests to access RFID events information), and results in a suitable set of access rules of the corresponding entity to the object(s). This way, the access policy can be dynamically reuse for any user that even had no relationship before. Second, fine grained policy access enforcement method to handle large amounts of RFID events information, using the created rules and contextual information, is presented. For evaluation purpose, a the proposed access control model

implementation with the spirit of inter-operability is developed and tested. This chapter fulfills the research goals number 1 and 5 which are stated in Section 1.7.1. The scientific contribution that constructs this chapter and is published in [6] as a journal publication.

Chapter 3 highlights our contributions in the open IoT/M2M middleware platform, along with our proposed secure access control mechanism. In this regard, both the architectural design of RFID middleware and secure access control approaches are employed to tackle the tag mobility and security issues. First, an architecture based on Session Initiation Protocol (SIP) signalling is proposed for the location and mobility management in the inter-enterprise RFID middleware sub-system. The proposed architecture is shown to greatly reduced the delay and is easily scale to many number of tags as compared with the existing system used as our reference point. Second, the access control mechanism, as one way to control the RFID events information to be accessed by other parties, is shown to integrate well with this approach. From the architectural point of view, performing access control at the SIP gateway is a better approach than at the EPCIS server because SIP gateway is the first entry point before the information stored at the EPCIS repository is accessed. This chapter fulfills the research goals number 2 and 5 which are stated in Section 1.7.1. The scientific contributions that constitute this chapter are published in several conference publications [2, 4]. Other contributions in the scope of ASPIRE project deliverables are also published [58, 57].

In Chapter 4, a secure access control model with authority delegation mechanism based on capability and contextual information is investigated and designed for IoT/M2M system. The key contributions of the proposed access control model are as follows. First, the scalability challenge in a highly distributed IoT/M2M system is addressed by introducing capability with identity of the Subject, i.e. an entity that request for an access of a certain resource. Secondly, the dynamic nature of IoT/M2M system that require a flexibility in access decision is tackled by incorporating contextual information in the access policies, access delegation by means of capability propagation, and introducing a virtual identity that is linked with a certain profile which can be flexibly defined by the user or device owner. Moreover, the privacy issue in access control is overcome by the virtual identity that is also linked with access policies. Third, the proposed access control model is shown to be secure against man-in-the-middle and replay attacks and able to achieve the authentication by introducing a nonce and some lightweight cryptographic operation. This chapter fulfills the research goals number 3 as stated in Section 1.7.1. The major scientific contributions that constitute this chapter are published in a conference publication [3] and a book chapter [5]. Other minor contributions

related to this chapter are also published in conference publication [42, 41]. An identity establishment and authentication method based on Elliptic Curve Cryptography (ECC) that complement the proposed access control model is presented in [42] and practical implementation along with the performance and security evaluation of the identity driven capability-based access control is demonstrated in [41].

Chapter 5 presents a practical implementation and use cases of a capability based access control for the IoT/M2M local cloud platform. The proposed capability access control in this chapter is aimed at solving the inherent issues, such as scalability and enforcing least privilege access principle, of the existing access controls, e.g. ACL, Role Based Access Control (RBAC) [53], and Attribute Based Access Control (ABAC) (e.g. Extensible Access Control Markup Language (XACML) [61]), by using capability. It also solves the access delegation issue in the existing IoT/M2M cloud platform that relies on API key and OAuth [46] for the access delgation, in which the OAuth application, i.e. the 3rd party application, is able to access all the resources tied with the delegating party under a specific access permission, e.g. Read, Write or Delete, by indicating a specific set of resource that can be accessed in the delegation. The issue of controling the delegation chain through capability propagation is also addressed by using a validation method based on a Public Key Infrastructure (PKI). In addition, the aforementioned validation method also improves the access delegation mechanism of the Capability-based Context-Aware Access Control (CCAAC) model in chapter 4, i.e. to make it more flexible and yet keeping the access delegation under control. Several uses cases show that the proposed capability based access control is suitable to the various operations of the IoT/M2M cloud platform. But in principal, it is also suitable for any kind of cloud computing platform. This chapter fulfills the research goals number 3 and partly 5 which are stated in Section 1.7.1. The scientific contribution that constitute this chapter is published in a conference publication [8].

In chapter 6, an analytical framework of an intrusion detection in access control system based on game theoretic approach is presented. Similar to chapter 5, the target platform being investigated in this study is the IoT/M2M local cloud platform which consists of a collection of distributed IoT/M2M gateways. Here, each gateway has a set of resources or assets with different security values whom the access is governed by the access control system that is integrated with an Intrusion Detection System (IDS). To model such system, a two-player multi-stage Bayesian game is proposed. The two players are attacker and defender, each of them has a constraint in terms of attack and monitor resource respectively. The Bayesian game is chosen due to some degree of uncertainties in the IDS and naturally the defender does not know whether

or not the attacker is malicious. The multi-stage type of game means the game is played repeatedly and the previous actions may influence the current decision, consequently the player, e.g. defender, needs to update its belief in each stage of the game and this is done by Bayes rule. Based on the proposed analytical framework, the set of Perfect Bayesian Equilibrium (PBE) strategies for both players in each stage game is formulated, and the framework is also analysed numerically in order to validate the proposed analytical framework. This chapter fulfills the research goals number 4 as stated in Section 1.7.1. The scientific contribution that constructs this chapter is published in a conference publication [7].

In chapter 7, the overall research work conclusions and future scope for the research work are contained. This ties up all the findings in our PhD study and enlightens the path for forthcoming researchers in this field.

To summarize, all the thesis chapters are correlated with each other as depicted in Figure 1.10.

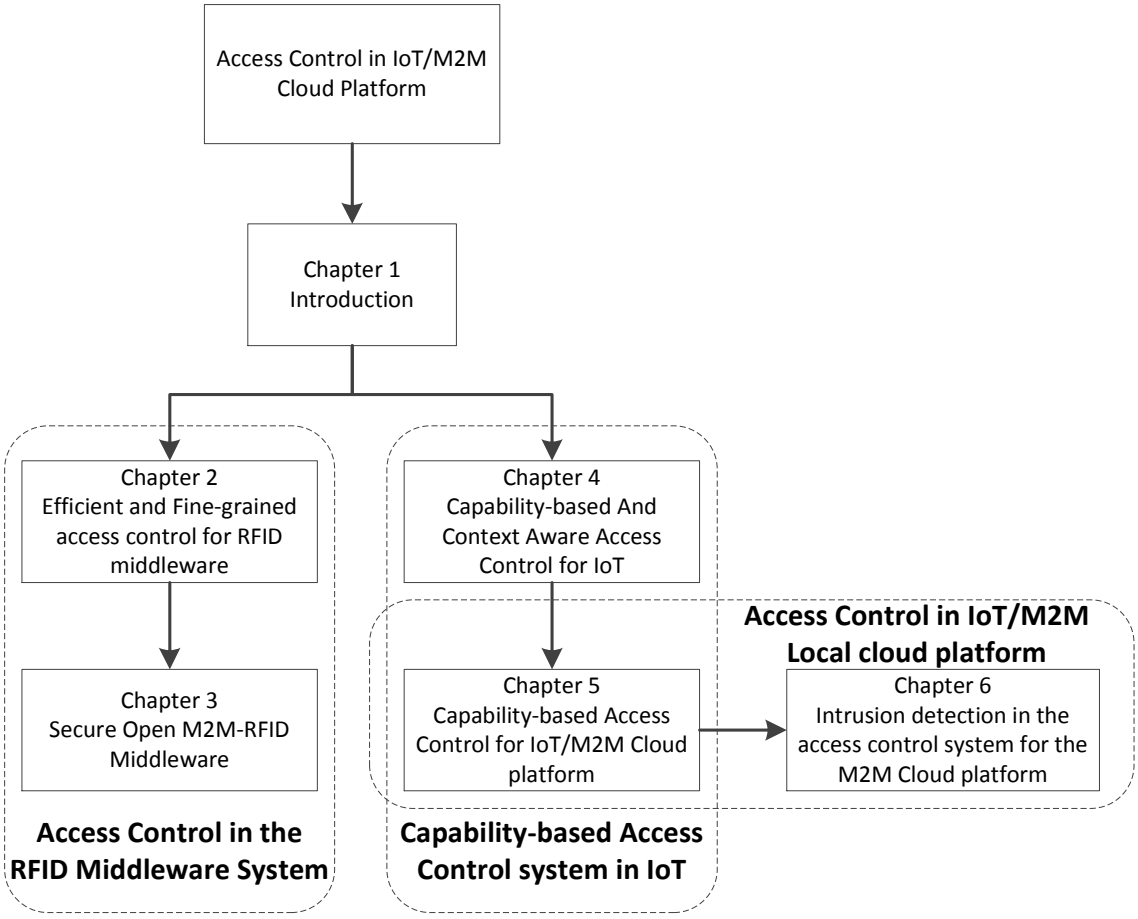

Figure 1.10: Correlation of dissertation chapters

1.9 References

[1] Julien Abeille, Mathilde Durvy, Jonathan Hui, and Stephen Dawson-Haggerty. Lightweight ipv6 stacks for smart objects: the experience of three independent and interoperable implementations. *Internet Protocol for Smart Objects (IPSO) Alliance, White Paper #2*, November 2008.

[2] Bayu Anggorojati, Kamil Çetin, Albena Mihovska, and Neeli R. Prasad. Rfid added value sensing capabilities: European advances in integrated rfid-wsn middleware. In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, June 2010.

[3] Bayu Anggorojati, Parikshit N. Mahalle, Neeli R. Prasad, and Ramjee Prasad. Capability-based access control delegation model on the federated iot network. In *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, September 2012.

[4] Bayu Anggorojati, Parikshit N. Mahalle, Neeli R. Prasad, and Ramjee Prasad. Efficient and scalable location and mobility management of epc-global rfid system. In *Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on*, June 2013.

[5] Bayu Anggorojati, Parikshit N. Mahalle, Neeli R. Prasad, and Ramjee Prasad. Secure access control and authority delegation based on capability and context awareness for federated iot. In Fabrice Theoleyre and Ai-Chun Pang, editors, *Internet of Things and M2M Communications*. River Publisher, 2013.

[6] Bayu Anggorojati, Neeli R. Prasad, and Ramjee Prasad. Efficient fine grained access control for rfid inter-enterprise system. *Journal of Cyber Security and Mobility*, 2(3 & 4), 2013.

[7] Bayu Anggorojati, Neeli R. Prasad, and Ramjee Prasad. An intrusion detection game in access control system for the m2m local cloud platform. In *The 19th Asia Pasific Conference on Communications (APCC) 2013*, August 2013.

[8] Bayu Anggorojati, Neeli R. Prasad, and Ramjee Prasad. Secure capability-based access control in the m2m local cloud platform. In *Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2014 IEEE 4th International Conference on*, May 2014.

[9] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787 – 2805, 2010.

[10] Axeda. M2m cloud service. *http://www.axeda.com/platform*.

[11] Guangdong Bai, Lin Yan, Liang Gu, Yao Guo, and Xiangqun Chen. Context-aware usage control for web of things. *Security and Communication Networks*, 2012.

[12] BETaaS. Building the environment for the things as a service. *http://www.betaas.eu*.

[13] Hyuntae Cho, Jongdeok Kim, and Yunju Baek. Large-scale active rfid system utilizing zigbee networks. *Consumer Electronics, IEEE Transactions on*, 57(2):379 –385, may 2011.

[14] J.P. Conti. The internet of things. *Communications Engineer*, 4(6):20 –25, dec.-jan. 2006.

[15] The National Intelligence Council. Disruptive civil technologies - six technologies with potential impacts on us interests out to 2025. In *Conference Report CR 2008-07*, 2008.

[16] Adam Dunkels and JP Vasseur. Ip for smart objects. *Internet Protocol for Smart Objects (IPSO) Alliance, White Paper #1*, September 2008.

[17] Glenn Brunette et al. Security guidance for critical areas of focus in cloud computing v2.1. *Cloud Security Alliance*, December 2009.

[18] ETSI. Machine-to-machine communications (m2m); functional architecture. Technical report, European Telecommunications Standards Institute (ETSI), 2011.

[19] Neil Gershenfeld, Raffi Krikorian, and Danny Cohen. The internet of things. *Scientific American*, 291(4):76 – 81, October 2004.

[20] S. Gusmeroli, S. Piccione, and D. Rotondi. Iot access control issues: A capability based approach. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, July 2012.

[21] S. Gusmeroli, S. Piccione, and D. Rotondi. A capability-based security approach to manage access control in the internet of things. *Mathematical and Computer Modelling*, 58(5-6):1189 – 1205, 2013.

[22] http://wiki.aspire.ow2.org.

[23] Jonathan Hui, David Culler, and Samita Chakrabarti. 6lowpan: Incorporating ieee 802.15.4 into the ip architecture. *Internet Protocol for Smart Objects (IPSO) Alliance, White Paper #3*, January 2009.

[24] Kai Hwang and Deyi Li. Trusted cloud computing with secure resources and data coloring. *Internet Computing, IEEE*, 14(5):14 –22, sept.-oct. 2010.

[25] RFID in Japan. Battery-less sensor tags. *http://ubiks.net/local/blog/jmt/archives3/005338.html*.

[26] EPCglobal Inc. Epc information services (epcis) version 1.0.1 specification. September 2007.

[27] EPCglobal Inc. Epcglobal object name service (ons) version 1.0.1. May 2008.

[28] EPCglobal Inc. The epcglobal architecture framework. December 2010.

[29] EPCglobal Inc. Gs1 epc tag data standard 1.6 - ratified standard. September 2011.

[30] EPCglobal Inc. Gs1 epcglobal tag data translation (tdt) 1.6 - ratified standard. October 2011.

[31] IoT-A. Internet of things - architecture. *http://www.iot-a.eu*.

[32] Wayne Jansen and Timothy Grance. Guidelines on security and privacy in public cloud computing. *NIST Special Publication*, December 2011.

[33] A.H. Karp. Authorization-based access control for the services oriented architecture. In *Creating, Connecting and Collaborating through Computing, 2006. C5 '06. The Fourth International Conference on*, 2006.

[34] Tom Karygiannis, Bernard Eydt, Greg Barber, Lynn Bunn, and Ted Phillips. Guidelines for securing radio frequency identification (rfid) systems - recommendations of the national institute of standards and technology. *NIST Special Publication*, April 2007.

[35] Hermann Kopetz. Internet of things. In *Real-Time Systems*, Real-Time Systems Series, pages 307–323. Springer US, 2011.

[36] K. Kumar and Yung-Hsiang Lu. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51 –56, april 2010.

[37] G. Lawton. Machine-to-machine technology gears up for growth. *Computer*, 37(9):12 – 15, sept. 2004.

[38] Libelium. Wireless sensor network with waspmote and meshlium. *http://www.libelium.com/development/waspmote/documentation*, 2013.

[39] LogMeIn. Logmein - access, manage and support computers remotely. *https://secure.logmein.com*.

[40] ITU-T FG M2M. Deliverable d2.1 - m2m service layer: Requirements and architectural framework. Technical report, ITU-T, 2013.

[41] Parikshit N. Mahalle, Bayu Anggorojati, Neeli R. Prasad, and Ramjee Prasad. Identity driven capability based access control (icac) scheme for the internet of things. In *6th IEEE International Conference on Advanced Networks and Telecommunication Systems (ANTS) 2012*, December 2012.

[42] Parikshit N. Mahalle, Bayu Anggorojati, Neeli R. Prasad, and Ramjee Prasad. Identity establishment and capability based access control (iecac) scheme for internet of things. In *Wireless Personal Multimedia Communications (WPMC), 15th International Symposium on*, September 2012.

[43] Friedemann Mattern and Christian Floerkemeier. From the internet of computers to the internet of things. In *From Active Data Management to Event-Based Systems and More*, volume 6462 of *Lecture Notes in Computer Science*, pages 242–259. Springer Berlin / Heidelberg, 2010.

[44] Enzo Mingozzi, Carlo Vallati, Giacomo Tanganelli, Giovanni Iovino, Valerio Di Gregorio, and Bayu Anggorojati. D1.4.1 - taas reference model. Technical report, BETaaS, 2012.

[45] Aikaterini Mitrokotsa and Christos Douligeris. Integrated rfid and sensor networks: Architectures and applications. In *RFID and Sensor Networks: Architectures, Protocols, Security, and Integrations*, Wireless Networks and Mobile Communications, pages 511–535. CRC Press, November 2009.

[46] OAuth. Oauth 2.0. *http://oauth.net*.

[47] R. Prasad. *My personal Adaptive Global NET (MAGNET)*. Signals and Communication Technology Book. Springer Netherlands, 2010.

[48] Mirko Presser and Alexander Ghulak. The internet of things: Connecting the real world with the digital world. *EURESCOM mess@ge - The Magazine for Telecom Insiders*, 2, september 2009.

[49] Intel WISP Project. http://seattle.intel-research.net/wisp/.

[50] B.P. Rimal, Eunmi Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*, aug. 2009.

[51] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266 – 2279, 2013.

[52] Antonio G. Ruzzelli, Raja Jurdak, and Gregory M.P. O'Hare. the rfid wake-up impulse for multi-hop sensor networks. In *1st ACM Workshop on Convergence of RFID and Wireless Sensor Networks and their Applications (SenseID) at the Fifth ACM Conference on Embedded Networked Sensor Systems (ACM SenSys 2007)*, November 2007.

[53] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *Computer*, 29(2):38 –47, feb 1996.

[54] Sanjay Sarma, David L. Brock, and Kevin Ashton. The networked physical world, proposals for engineering the next generation of computing, commerce & automatic-identification. *MIT-AUTOIDWH-001*, 2000.

[55] Zach Shelby and Carsten Bormann. *6LoWPAN: The Wireless Embedded Internet*. Wiley Publishing, 2010.

[56] Zach Shelby, Klaus Hartke, Carsten Bormann, and Brian Frank. Constrained application protocol (coap). Technical report, IETF, 2012.

[57] John Soldatos, Nikos Kefalakis, Nektarios Leontiadis, Bayu Anggorojati, Simone Frattasi, Neeli Prasad, Didier Donsez, Gabriel Pedraza, Kiev Gama, and Jacky Estublier. D3.5: End-to-end infrastructure management. Technical report, ASPIRE, 2011.

[58] John Soldatos, Nikos Kefalakis, Nektarios Leontiadis, Nikolaos Konstantinou, Nathalie Mitton, Loc Schmidt, Roudy Dagher, Mathieu David, Bayu Anggorojati, Simone Frattasi, Neeli Prasad, Didier Donsez, Gabriel Pedraza, and Kiev Gama. D3.4b: Core aspire middleware infrastructure. Technical report, ASPIRE, 2010.

[59] Rolf H. Weber. Internet of things – new security and privacy challenges. *Computer Law & Security Review*, 26(1):23 – 30, 2010.

[60] Geng Wu, S. Talwar, K. Johnsson, N. Himayat, and K.D. Johnson. M2m: From mobile to embedded internet. *Communications Magazine, IEEE*, 49(4):36 –43, april 2011.

[61] XACML. https://www.oasis-open.org/standards#xacmlv2.0.

[62] Xively. Api docs - security.

[63] Xively. What is xively ? *https://xively.com/whats_xively*.

[64] Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian, and Aoying Zhou. Security and privacy in cloud computing: A survey. In *Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference on*, November 2010.

2

Efficient Fine-Grained Access Control for RFID Inter-Enterprise Middleware System

Access control management is a very challenging task in an inter-enterprise RFID system due to huge amounts of information about things or objects that can be collected and accessed to and from the system. Furthermore, the information stored in the inter-enterprise RFID system contains sensitive and confidential data related to the activities of the organization involved around the RFID system. Hence, the efficiency and high-granularity are critical in the design of access control for such system. This chapter presents a novel access control model which is efficient and fine grained for such a system. A detail definition and mechanism of the access control model are described in the chapter. A system implementation is developed for the evaluation purpose. An important performance measure in big data processing is delay in processing time, thus the evaluation aims at measuring the access control processing time. The evaluation results show that the model is consistent, and is able to achieve less delay than the inter-enterprise RFID system without access control at a certain point.

2.1 Introduction

RFID technology allows the everyday things to be interconnected to the internet world, thus the key component towards the full deployment of the IoT vision [2]. From the system components and architecture perspectives, RFID system employed by any organization in its activity may consist of three sub-systems, namely RF, enterprise, and inter-enterprise sub-system [11]. Inter-enterprise sub-system in particular, is the most important component that enables the objects or things visibility and tracking throughout their life cycle, i.e. in supply chain industry, etc. When the information about thousands of things or objects is able to be gathered and accessed, consequently access control management of such information is a great challenge.

The technical specification, including the standard interfaces and data format, that enables the inter-enterprise information sharing of RFID events data is specified in the EPCIS specification [8] issued by EPCglobal. The EPCIS repository, i.e. software implementation of this specification, in particular aims at receiving application-agnostic RFID data, translate it into a corresponding business events (e.g. business process, business location, event time, etc), and then make the events available and accessible by upstream applications. Since the EPCIS repository potentially contains sensitive and confidential data of any individual or organization, the access to such information through its interfaces needs to be managed properly. In this regard, it is important to mention that the access control mechanism in the EPCIS specification is left open to each specific implementation. Additionally, efficiency, fine level of granularity, and trust are important keys to the access control design since highly dynamic and huge amount of events data is expected to be generated by potentially thousands of tagged objects which are of any interest for individuals or organizations that have even had any relationship before.

There are two main contributions of this chapter. First, a dynamic and efficient access policy mechanism of an object or group of objects, based on the attributes and vocabularies of EPCIS is introduced. This access policy takes the profile of the accessing entity (i.e. individual or organization that requests to access RFID events information – this term will shortly be referred as *user* throughout the rest of this chapter), and results in a suitable set of access rule of the corresponding entity to the object(s). This way, the access policy can be dynamically reused for any user that even had no relationship before. Second, fine grained policy access enforcement method to handle large amounts of RFID events information, using the created rules and contextual information, is presented. For evaluation purpose, a system implementation of the proposed access control model is developed, tested, and verified.

The remainder of this chapter is organized as follows: An overview of related works in access control is given in Section 2.2. The problems, requirements, and realistic assumptions along with a real life example for designing an access control framework in RFID is described in Section 2.3. The proposed access control framework, along with the definition of access policy of the object(s), mechanism to generate access rules for the accessing entity, and the access policy enforcement mechanism, is explained in Section 2.4. The system implementation of the proposed access control model is presented in section 2.5. The evaluation results and findings are discussed in Section 2.6. Some qualitative discussion regarding important features of secure system and access control constituted in the proposed model as well as comparison with existing access control model are presented in Section 2.7. Finally, the conclusion and future directions of this work are given in Section 2.8.

2.2 Related works

Study in various types of access control models have been quite well established within the computing and information technology field. Our particular interests are in incorporating the contextual information and dynamically create access rules based on a pre-defined set of policies. XACML [13] is an eXtensible Markup Language (XML) framework to describe access control policies for web based resources. The XACML specification incorporates some contextual information into access decisions, but it has no formal context-aware access control model. In addition, the access decision from the evaluated policies in XACML is only limited to four pre-defined categories, i.e. Permit, Deny, NotApplicable, or Indeterminate, which greatly reduces the granularity of access decision results.

RBAC [12] is an access control model that is widely used and further derived into different models, due to its suitability in almost any organization which consists of different roles with some levels of hierarchy. Temporal aspects of RBAC were addressed in Temporal RBAC (TRBAC) [3], which focuses on temporal availability and dependency of roles. General Temporal RBAC (GTRBAC) [10] is an extension of TRBAC model that is capable in expressing a wide range of temporal constraints – in particular time periodicity as well as duration, and de/activating as well as enabling constraints – on roles. An XML specification of GTRBAC has been introduced in [5] and the extension of X-GTRBAC which incorporates trust in assigning roles to users has been presented in [4]. Although these models have a detailed definition of context information but the role based model, i.e. with user-to-role

and role-to-permission mapping, does not fit with the requirement of RFID inter-enterprise system.

CCAAC [1], another type of access control model that supports contextual information and is based on capability. In addition, CCAAC provides a framework where a valid capability as a mean for an access request to be granted, is created for any user based set of access policies attached to an object or group of objects. Here, *object* refers to resource to be accessed by any user. The CCAAC offers efficient and dynamic way of managing access control through the evaluation of user's profile and contextual information via the corresponding access policies upon the capability request to certain object(s), which is important when dealing with huge numbers of objects and users simultaneously, e.g. in IoT or RFID system. Moreover, it also supports access delegation and revocation. However, the type of action and access decision result limits the level granularity, and the context information is not formally modelled.

A fine grained access enforcement specially designed for the EPCIS events data through a rule-based policy language for Auto-ID events, called Auto-ID Authorization Language (AAL), has been introduced in [6]. In addition, an efficient policy enforcement mechanism and implementation based on Structured Query Language (SQL) query rewriting was presented. The main drawback of AAL as presented in [6] is that the access policy is manually assigned to users which is impractical in the real life situation. The dynamic generation, assignment, and revocation of access policies were not considered as well.

2.3 Problem descriptions and the system requirements

The RFID events data in the EPCIS repository, which is known as EPCIS events, is categorized into four types of events namely *ObjectEvent*, *AggregationEvent*, *QuantityEvent*, and *TransactionEvent* [8]. Each of them describes different type of event taking place in relation to the RFID tag, which is represented by EPC ID, in the business process within the company. In addition, two types of data, i.e. RFID application-agnostic and master/company data, are comprised in the EPCIS repository. Here, the master data, e.g. eventTime, action, bizStep, etc, provides some necessary business context to interpret the EPCIS events [8].

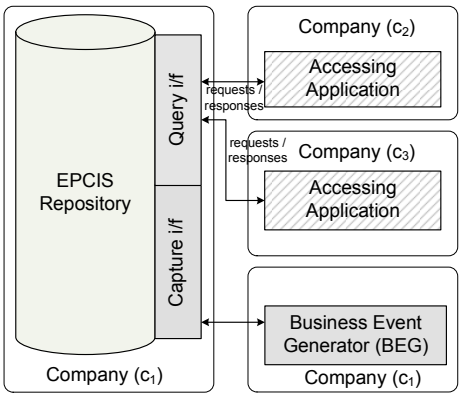

Figure 2.1: The interactions of EPCIS repository interfaces in an inter-enterprise RFID system

A simplified example of an inter-enterprise RFID system deployment involving an EPCIS repository is depicted in Fig. 2.1. In this example, the EPCIS repository is owned by a company c_1, and is accessed by companies c_2 and c_3. The RFID events consisting of RFID and master data generated by a BEG module owned by c_1 is captured via capture interface and stored in the EPCIS repository as EPCIS events. The EPCIS events stored in EPCIS repository can then be accessed by other companies through the query interface. According to a comprehensive definition and description of the EPCIS specification [8], the EPCIS events can be interpreted and gives a valuable information regarding the business activities of company c_1 by business context information and RFID data. For example, the interpreted EPCIS event can give a figure about production volume, sales activity of certain products, inventory status, etc. Obviously, these are very sensitive information that a company want to reveal as minimum as possible to other parties by managing access to EPCIS event data in high level of granularity. In addition, the the RFID tags' owner may want to restrict the access to particular information related to the activity of the tags which might reveal another type of sensitive information apart from the business related information.

Based on the identified problems, the following requirements for access control in an inter-enterprise RFID system should be foreseen:

- **Context-awareness:** The access control system should be design to support rich business context information that are contained in the EPCIS events.
- **Dynamic rule:** Providing a fine grained access control in a highly dynamic EPCIS events data that is generated continuously, it is almost

impossible to assign a static access rights for particular users to certain part of data or attributes. Therefore, dynamic access rule should be generated based on the specified rules in the access policy and the requested set of information. In addition, the policy should support flexible inclusion of new events.

- **Dynamic access assignment:** There are certainly various types of users that are trying to gain access to the EPCIS events data which are probably not known before by the EPCIS repository's system administrator, i.e. the responsible person to create the access policy. Hence, a dynamic mechanism to assign access rights to users is also required.
- **Object based policy:** The access policy based on particular RFID tag IDs, e.g. on the Object Class or serial numbers level, is also required to restrict the access to information related to tag's activities by the tag's owner.

2.4 Access control model

2.4.1 Assumptions

Based on the identified problems and how the inter-enterprise RFID system operates, the following assumptions are made as a baseline to design an efficient, dynamic, fine grained access control for inter-enterprise RFID system:

- The authentication phase has been carried out before the access control process takes place.
- The EPCIS events stored in the EPCIS repository are only events generated by the EPCIS repository's owner, e.g. a company.
- The set of contextual information, i.e. attributes of EPCIS events, such as bizStep, action, disposition, readPoint, epcList, etc, and their values are known to the EPCIS repository's owner. The values of some attributes are always fixed, e.g. $action = \{'ADD', 'OBSERVE', 'DELETE'\}$, while the values of attributes like *epcList* are dynamics but the company has a full knowledge of all EPC IDs involved in their business transactions.
- The EPCIS owner may or may not know the users that are accessing its EPCIS events through the query interface. But the user's profile, such

as company name, location, business area, etc, can be obtained through a trusted means, e.g. via trusted third party organization.

- For each query of EPCIS events, the user may optionally specify particular Event Type(s) and some contexts or event attributes, e.g. eventTime, action, bizStep, etc, as stated in the EPCIS specification [8]. It is important to note that the default query operation according to the EPCIS specification is that all the available information will be returned unless specified otherwise in the query.
- The RFID tag's owners have the knowledge of their tags' IDs in order to create access policies for an individual tag or a set of tags.

These assumptions lead us to propose an access control model that will be explained in the following subsections.

2.4.2 Definitions

2.4.2.1 Elements, Attributes, and Values

First of all, the data structure in our proposed access control framework is based upon $Element \rightarrow Attribute \rightarrow Value$ ternary relationship which maps each element to its attributes and their values. *Element* is a set of elements which could be user profile, P, or EPCIS event, E. Each of the *element* consists of several attributes and each of the attribute can have a set of possible values.

2.4.2.2 EPCIS Event

Let E_i be the i^{th} EPCIS event within the set of EPCIS events element. As mentioned previously, according to [8] E_i could be an *ObjectEvent*, *AggregationEvent*, *QuantityEvent*, or *TransactionEvent*. Each of E_i consists of a set of attributes as defined in details in [8]. Let us call the j^{th} attribute of an E_i as AE_j. Here, the relationship between E_i and AE_j can be expressed in the following notation: $E_i = \{AE_1, \cdots, AE_n\}$.

The value of each AE_j, let us call it as VE_{jk}, can either be a single value or a list of values, e.g. in the case of the *epcList*. Moreover, the value of some attributes may be among some pre-defined values. In any case, the relationship between AE_j and VE_k can be generally expressed as $AE_j = \{VE_{j1}, \cdots, VE_{jn}\}$.

2.4.2.3 User profile

In addition to our previous assumption, the administrator can define a set of user profile based on several attributes and values. Now, let P_i be the i^{th} user profile among a set of user profiles defined by the administrator. Similar to E_i, each P_i consists of several user profile attributes, e.g. AP_j, and each AP_j may consists of a set of values, i.e. VAP_{jk}, defined by the administrator.

$$AP_j = \{VAP_{j1}, \cdots, VAP_{jn}\}$$

2.4.2.4 Objects

The *Object* field can be expressed as a single tag ID or as a pattern which can apply to a set of tags, according to the TDS specification [9]. In mathematical form, the *Object* O can simply be expressed as a set of object values: $O = \{VO_i, \cdots, VO_n\}$

2.4.2.5 Condition

Condition is the key component to provide dynamic rule and fine-grained access control. It is important to mention that the *condition* in this proposed model is not a requirement for granting an access or not. Rather, it is defined as a set of constraint applied to the contextual information. In our case, the contextual information is specific to the business contexts or the EPCIS event attributes (AE_j).

The way the condition is expressed in the proposed access control policy follows the white-listing principle, meaning that the access to information that is not explicitly mentioned in the condition is not allowed, which is the opposite of the nature of the EPCIS query specification as explained earlier. This principle is valid in general, and especially in our case where the EPCIS event attributes are defined in great details.

With this principle in mind, a condition can be expressed in general as subset of AE_j, i.e. $AE_j^s \subseteq AE_j$, with respect to a set of all possible values of a particular AE_j. This implies that a condition can set whether to allow all, partly, or none of the values to be accessible. For example, let say an EPCIS event attribute consists some pre-defined values, e.g. $AE_j = \{a, b, c\}$. Some possible condition for such an attribute AE_j are $AE_j^s = AE_j = \{a, b, c\}$, $AE_j^s = \{b, c\}$, or $AE_j^s = \varnothing$, which means that the values of AE_j is shown fully,

partially (e.g. only $\{b, c\}$), and none. This way of describing the condition is also to fulfil the purpose of maintaining the consistency upon the existence of multiple rules.

On a practical stand point in writing a condition, there are two possible ways to specify it: first, by explicitly stating what set of values, i.e. VE_{jk} are accessible; second, by describing values in certain ranges using a comparison operator. The first way is more static and suitable for attributes that have some pre-defined values, e.g. action, business location, disposition, etc, whereas the later one is more dynamic and capable of specifying a condition for event that has not happened yet, such as the *eventTime*. In such a case, general statement of the condition of the rule i with respect to AE_j^s is as follows:

$$C_i = AE_{i1}^s \wedge \cdots \wedge AE_{ij}^s \wedge \cdots \wedge AE_{in}^s \tag{2.1}$$

2.4.2.6 Rule

A *rule* consists of a set of *conditions*, C and an EPCIS event type, ET. General representation of a rule using a **if-then** relationship is: $ET \Rightarrow C$. A set of rules together with a user profile P or a set of objects O forms a policy that will be explained in the next Subsection (2.4.2.7).

2.4.2.7 Policy

In the proposed model, there are two types of *policies*, namely user based and object based policies. A *user based policy* consists of a set of *profiles* P and a set of rules, while an *subject based policy* consists of a set of *objects* O and a set of rules. A set of different rules that are applicable for a *request*, i.e. matched to the *request*'s P or O, and having the same ET value are combined into a new applicable rule using a *disjunction* operation after resolving all the conflicting policies. In addition, multiple policies may also be applicable to an access *request*, thus a policy combining mechanism should take place in order to create a final applicable rule for an access *request*. Finally, the final applicable rule would then be evaluated with the *conditions* presented within the access *request* in order to get a final access decision. Please note that the access decision is not in a form of static *permit* or *deny*, rather it is in a set of access constraints or conditions that explicitly authorize which data can be accessed by the user. The whole mechanisms of the rule and policy combining, and then evaluate them with the access *request* are explained in great details in the Subsection 2.4.3.

2.4.2.8 Request

An access *request* must consists of a *profile* P, and optionally a set of EPCIS event types ET and a set of *conditions* C. Both ET and C are optional fields in a *request* due to the nature of the EPCIS query specification. For a *request* to be evaluated with a certain policy, the *request's* profile P_{REQ} should match with the *policy's* profile P_{pol}. If a single or set of ET(s) are specified in the *request*, then only rules that matched with the specified ETs are applicable, otherwise all rules in the policy are applicable for the *request*.

2.4.3 Access rules evaluation

Access rules evaluation is at the heart of the whole model in order to provide a fine-grained access control. The results of this evaluation is a final access decision in a form of a set of explicit access constraints.

There are two important steps in the access rules evaluation. First, all applicable rules are combined by using union operation. Second, the combined rules' condition is then evaluated against the condition presented by access request using an intersection operation. The general expression of this evaluation is depicted as follows:

$$C^{REQ} \cap (C_1 \cup \cdots \cup C_p) \tag{2.2}$$

where:

- p: number of matched access rules.
- $C_i = (AE^s_{ij} \wedge \cdots \wedge AE^s_{in})$: a set of conditions specified in the i^{th} matched access rule.
- $C^{REQ} = \{AE^r_k, \cdots, AE^r_m\}$: a set of EPCIS attributes conditions specified in user request.

The access rules evaluation as expressed in (2.2) is obtained through some steps illustrated in the pseudo-code 1.

The procedure starts by combining all conflict-free conditions of the applicable rules using union set operation. Once a combined rule is obtained, the access rule evaluation is started by grouping the similar event type attribute, i.e. AE pair, from between the C^{REQ} and the $CombineRules$, and then a set

Algorithm 1 access rules evaluation procedures

procedure RULESEVALUATION(C^{REQ}, SC)
  for all AE^s_{ij} in C_i in SC **do**
    for all C_i in SC **do**
      $AE^{s'}_j = UnionAll(AE^s_{xj})$
    end for
    $CombineRules = Conjunction(AE^{s'}_j)$
  end for
  $Condition =$ ""
  for all $AE^{s'}_j$ in $CombineRules$ **do**
    if $Condition \neq$ "" **then**
      $Condition \mathrel{+}= \wedge$
    end if
    for all AE^r_k in C^{REQ} **do**
      if $j == k$ **then**
        $Condition \mathrel{+}= AE^{s'}_j \cap AE^r_k$
      end if
    end for
    if notFound($k == j$) **then**
      $Condition \mathrel{+}= AE^s_j$
    end if
  end for
end procedure

intersection operation is applied, i.e. $AE_j^{s'} \wedge AE_k^r$, where $j = k$. In case of the pair of $AE_j^{s'}$ is not specified in the C^{REQ}, then only the condition as specified in AE_j^s will be applied. At last, the final expression of access rule evaluation as depicted in (2.2) is obtained.

2.5 System Implementation

Two important components in this system implementation are the Policy repository as well as evaluation, and Access Policy enforcement components. In addition, the overall implementation is designed to be modular and complies with the EPCIS query interface specification. For this purpose, the proposed access control is implemented through web service interface and be part of the AspireRFID middleware [7] module. The AspireRFID middleware itself is one of the open source implementation of RFID middleware based on EPCglobal specifications.

2.5.1 System architecture

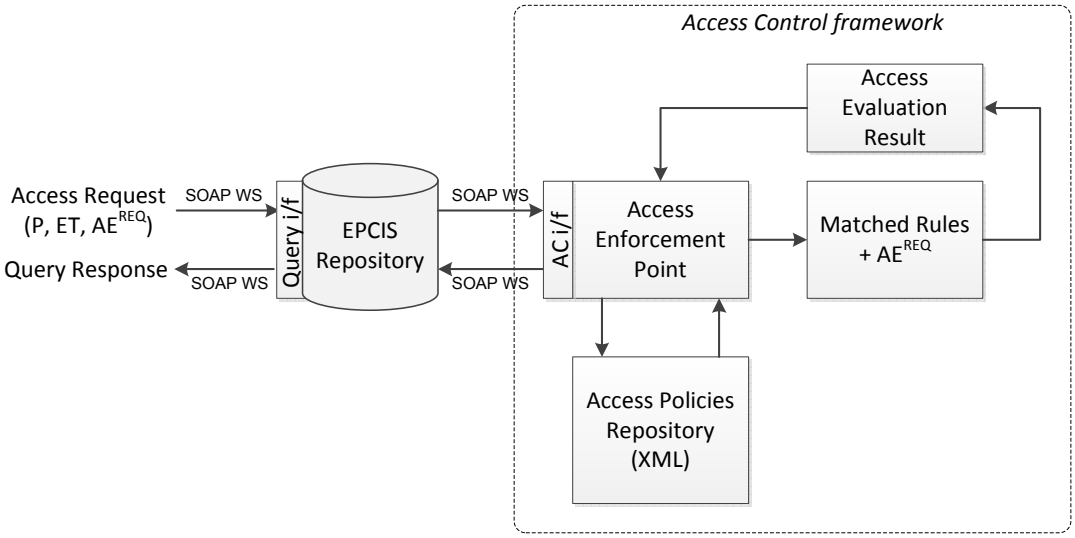

Figure 2.2: System architecture of the implemented access control model

The system architecture of the implemented access control model is depicted in Fig. 2.2. The Access Control Framework is communicating with the EPCIS repository through web service interface. Once the incoming query or access request is received, the access policy enforcement point will query

the Access Policies Repository to find the matching policies with the request according to the procedure explained in Section 2.4 earlier. Once a set of matching policies or rules is found, it will be evaluated against the access request. Finally, the result of the access control evaluation will be sent back to EPCIS repository through the Access Enforcement Point via the web interface.

The access policy enforcement is an important component in a policy based access control implementation after an access decision has been taken. In our implementation, the access control policy enforcement is implemented in a form of a modified *QueryParams*, i.e. the query parameters defined in the EPCIS specification. The advantage of this method is that the implementation complies fully with the EPCIS query interface specification while stay modular without a necessity to greatly modify the EPCIS server implementation. Moreover, it does not depend of the actual implementation of the database technology for the EPCIS repository. However, the main drawback is that this method does not fully leverage the proposed access control model due to the nature of *QueryParams* description in the EPCIS specification where the parameter that is not explicitly mentioned in the *QueryParams* will be included in the query results. Nevertheless, it still allows us to perform an evaluation over most of the important functionalities of the proposed access control model. Additionally, if the access decision return an empty result, i.e. no matching policy is found in the repository, the query results of the EPCIS query interface implementation will return a *SecurityExecption*.

2.5.2 XML specification of access control policy

The policy repository and evaluation component implementation uses XML and Java Architecture for XML Binding (JAXB) technologies. The proposed access control policy specification as explained in Section 2.4 is translated into a set of XML specification and is practically stored as an XML file. Thanks to the JAXB technology, the access control policies can be created as an XML file and the stored access control policies can be translated into Java Objects to be further evaluated.

The usage of XML for describing policies in the proposed access control model is motivated due to the fact that the XML has been well accepted and widely used in the heterogeneous IT enterprise environment across different platforms. Among others, XACML [13] is the most famous XML based access control system. Particularly in our system implementation, the XML is used it model the data structure of the proposed access policy elements as described

in Section 2.4. To give an overview of the XML specification for the proposed access control model, X-Grammar notion as introduced in [5] is used.

2.5.2.1 Profiles

To match the incoming user request with the suitable access control policy, the notion of user profile (*Profile* in short) is used. Different types of profiles are created by the system administrator based on several set of attribute-value pairs. The snippet of *Profile* XML representation in the X-Grammar, e.g. `XProfiles` is shown in Table 2.1.

Table 2.1: The grammar of of `XProfiles` in XML representation

<table>
<tr><td colspan="2"><!- - Profiles Definition> ::=
<Profiles>
{<!- - Profile Definition>}+
</Profiles></td></tr>
<tr><td><!- - Profile Definition> ::=
<Profile
ProfileId = (profile ID)>
<Attributes>
{<!- - Attribute Definition>}+
</Attributes>
</Profile></td><td><!- - Attribute Definition> ::=
<Attribute
AttrName = (name)
IsMandatory = ”true|false”
DataType = (type)>
(Attribute Value)
</Attribute></td></tr>
</table>

In Table 2.1, `ProfileId` is a unique *Profile* identifier to differentiate this profile with others. `IsMandatory` indicates whether or not the ***`Attribute`*** is a mandatory field. A set of `Attribute` along with the `IsMandatory` attribute in the `XProfile` provides a vocabulary to express the attributes needed by the users upon sending a *Request*.

2.5.2.2 Object

The *Object* element is used for creating access policy for a set of EPC IDs. It is also defined by the system administrator, in particular the tag's ”owner”. To express a set of objects in the XML representation, the EPC ID patterns format that is defined in the EPCGlobal TDS specification [9] is employed. The XML representation of the *Object* is referred as `XObjects`. The snippet of `XObjects`' grammar is depicted in Table 2.2.

Table 2.2: The grammar of `XObjects` in XML representation

```
<!- - Objets Definition> ::=
<Objects
   ObjId = (objects ID)>
  { <ObjectPattern> (EPC Pattern)  </ObjectPattern> }+
 </Object>
```

The `ObjId` is a unique group of objects identifier. According to [9], some examples of `ObjectPattern`'s value are "urn:epc:pat:gid-96:145.233.*" are "urn:epc:pat:gid-96:145.233.[350-360]". The first pattern example applies to any tag whose *General Manager Number* is 145, whose *Object Class* is 233, and whose *Serial Number* may be anything. The second example applies to any tag whose *General Manager Number* is 145, whose *Object Class* is 233, and whose *Serial Number* is between 350 and 360.

2.5.2.3 Event Attribute

The *Event Attribute* refers to the fields of *EPCISEvent* classes and the query parameters in the *SimpleEventQuery* that are defined in the EPCIS specification [8]. According to [8], the query parameters can be expressed by through any comparison operators, e.g. $=, <, \leq, >, \geq$, over a single value or a set of values. The XML specification of the *Event Attribute* in our proposed model needs capture such requirements to be compatible with the EPCIS specification. In the actual system deployment, the *Event Attributes* are part of the access request which are translated from the EPCIS query parameters. The grammar of *Event Attribute* XML representation – referred as `XEventAttribute` – is shown in Table 2.3.

Table 2.3: The grammar of `XEventAttribute` in XML representation

```
<!- - EventAttribute Definition> ::=
<EventAttribute
   AttrId = (Event Attribute ID)>
  <Operator>{eq|lt|leq|gt|geq}</Operator>
  { <AttrributeValue> (value)  </AttrributeValue> }+
 </EventAttribute>
```

The `AttrId` is a unique name of the EPCISEvent class's (and its subclasses) field as defined in [8]. The `Operator` is a comparison operator which can

either be equal, less-than, less-than-or-equal, greater-than, and greater-than-or-equal. It is particularly designed to support general expression of the query parameters in the *SimpleEventQuery* [8]. Finally, the `AttributeValue` can be one or more based on the specification of the query parameters where the parameter value can be a single value or a list of values. For instance, the `LT_eventTime` query parameter can be described by using *"lt"* `Operator` and only one `AttributeValue` in the XML language, but the `EQ_action` uses *"eq"* `Operator` and several `AttributeValues`.

2.5.2.4 Rule

A *rule* in the proposed access control policy consists of two mandatory elements, namely an *EventType* and a set of *Conditions*. The grammar of *Rule* XML representation, which we call it as `XRules`, is shown in Table 2.4.

Table 2.4: The grammar of `XRules` in XML representation

<!- - Rules Definition> ::=	<!- - Rule Definition> ::=
<Rules>	<Rule
{<!- - Rule Definition>}+	RuleId = (rule ID)>
</Rules>	<EventType>{OE|AE|QE|TE}</EventType>
	<Condition>
	{<!- - Logical Expression>}
	</Condition>
	</Rule>

`RuleId` is a unique identifier of the *Rule*. The value of `EventType` refers to either *ObjectEvent*, *AggregationEvent*, *QuantityEvent*, or *TransactionEvent* respectively. The `Condition` is expressed through *Logical Expression*, whose the grammar in XML language is defined in Table 2.5.

The *Logical Expression* (`LogicalExpr`) consists of a set of `Predicates`, in which all the `Predicates` are evaluated by a logical operator `op`: AND, OR, or NOT. The `LogicalExpr` can also be nested inside the `Predicate`, allowing the `Condition` to be expressed into different level of constraints. Apart from the nested `LogicalExpr`, each of the `Predicate` also comprises a comparison operator (`Operator`), a `ParamName`, and one or more `ParamValue`. Please note that the `Predicate`'s grammar is almost similar to `EventAttribute`, except that the `ParamName` is added as its element. This is because the proposed access control especially evaluates each `Predicate` within a `Condition` against the corresponding `EventAttribute` constraint that is part of the access request.

Table 2.5: The grammar of Logical Expression in XML representation

<!- - Logical Expression> ::=
`<LogicalExpr op =` {AND|OR|NOT}`>`
{<!- - Predicate>}+
`</LogicalExpr>`
<!- - Predicate> ::=
`<Predicate>`
[{`<Operator>`{eq|lt|leq|gt|geq}`</Operator>`
`<ParamName>`(parameter name)`</ParamName>`
{`<ParamValue>`(value)`</ParamValue>`}+ }
< !- - LogicalExpression>]
`</Predicate>`

2.5.2.5 Policy

The grammar of `XPolicies` in XML language is shown in Table 2.6. As per the definition of *Policy* in Section 2.4.2.7, a *Policy* may contain a set of `Profiles`, i.e. subject based policy, or a set of `Objects`, i.e. object based policy. In any case, any `Policy` should contain a set of `Rules` as the main component to evaluate the access request.

Table 2.6: The grammar of `XPolicies` in XML representation

<!- - Policies Definition> ::=	<!- - Policy Definition> ::=
`<Policies>`	`<Policy`
{<!- - Policy Definition>}+	`PolicyId =` (policy ID)`>`
`</Policies>`	[<!- - Profiles Definition>]
	[<!- - Object Definition>]
	<!- - Rules Definition>
	`</Policy>`

2.6 Evaluation results and analysis

A series of experiments or evaluation have been carried out based on our implementation in order to measure the performance of the proposed access control model. The main purposes of the evaluation are to validate the functionality of the proposed access control model and to measure the performance in terms of delay time.

2.6.1 Evaluation procedures

For the testing purpose, one policy for a particular user profile is prepared. The policy consists of different rules with several conditions for different EPCIS event types. It is important to note that since generating EPCIS events already requires some complex procedures, we focuses on generating only one type of business event in the test scenario. Hence, the generated EPCIS events are always having fixed event attributes values (VE_ij), except for the *eventTime*, *recordTime* and EPC ID. As a result, the final access control enforcement mainly depends on conditions related to *eventTime* and EPC ID parameters or attributes in the practical experiments.

Regarding the first objective of this evaluation, it is shown that the access control works as it should be. The query results only returns the EPCIS event data that fulfil the conditions set in the policy and the final access decision. On the other hand, a *SecurityException* will be returned if there is no matching policy found in the access policy repository.

Concerning the second objective of the evaluation, the delay performance of the proposed access control model is compared against the EPCIS system without any access control applied. For this purpose, we have tested our proposed access control model on a system with Intel Core i7 2.80 GHz CPU, 8 GB RAM, running Windows 7, and Java 6.31. In addition, an Apache Tomcat 6.35 server is used to deploy all the AspireRFID Middleware modules as well as our access control module, and a MySQL 5.2 Server is used as the EPCIS events data repository. Although the measurement results strongly dependant on the implementation, nevertheless we can expect to draw some qualitative conclusion out of the results and further improve the system implementation if necessary.

Two measurement scenarios are carried out to observe the delay performance and behaviour of the implemented proposed access control model. The delay is defined as the time consumed between the query request being sent and received by the EPCIS query client, i.e. $T_{received} - T_{sent}$. For each scenario, the same measurement point is done repeatedly for 700 times. For the purpose of explanation in the rest of this chapter, we will refer the case when the access control is used as the *first case* and the *second case* refers to the case when access control is not used.

2.6.2 The impact of varying the number of read tags

The first scenario aims to observe the delay time behaviour when the number of read tags in each event stored in the EPCIS repository is varied. In this scenario, the number of EPCIS events are fixed to three events at the EPCIS repository. It should be noted that the query results of the *first case* always returns one EPCIS events, while the *second case* always returns all three events since no access control is applied.

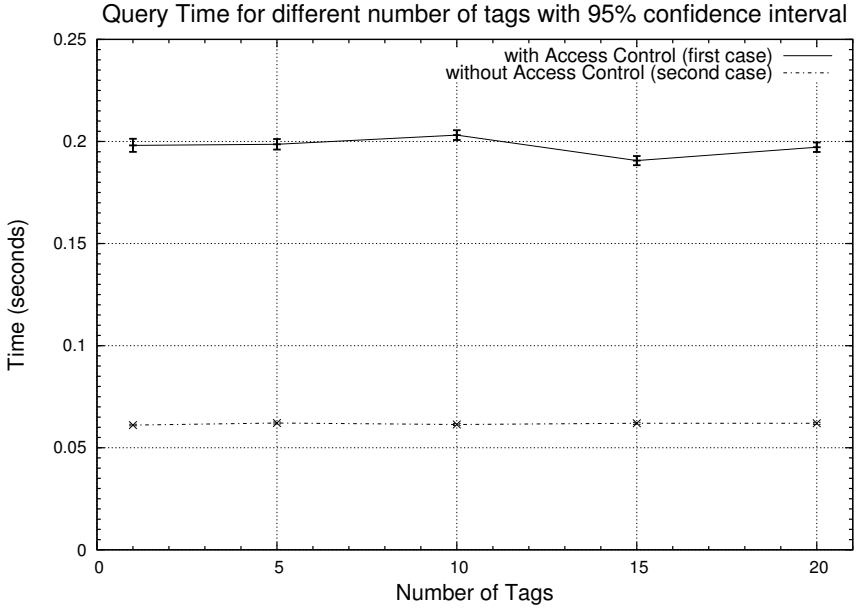

Figure 2.3: Query delay time for different number of tags with 95% confidence interval

There are two important findings that can be derived based on the results of this scenario which is shown in Fig. 2.3. First, the average delay of the *first case* is three times higher than that of the *second case*. This finding is quite expected because it involves extensive XML processing, e.g. creating and parsing of Simple Object Access Protocol (SOAP) message and parsing the access policy, which is time consuming. The delay performance could be improved in general through a more tightly coupled implementation with the expense of sacrificing the flexibility and modularity of such an open system. Second, the EPCIS query delay time for both cases is relatively constant regardless of the number of tags read in each EPCIS event. Although some small variation is shown in the *first case*, the confidence interval margin is rather high and it could be due to the inconsistency of web service invocation of access control service. Thus it is quite safe to neglect the small variation. Nevertheless, the second finding is

quite interesting because the amount of data queried from the database would contribute to the query time delay.

2.6.3 The impact of the number of the EPCIS events data in the repository

Based on the second finding in the first scenario, we would like to check the impact of varying the number of EPCIS events in the repository to the query time delay. Knowing that the number of tags does not change the delay behaviour, the number of tag included in the EPCIS events stored within the repository is fixed to only one tag in this scenario. Similar to the first scenario, the *first case* always returns one EPCIS events while the *second case* always returns all the EPCIS events available in the repository.

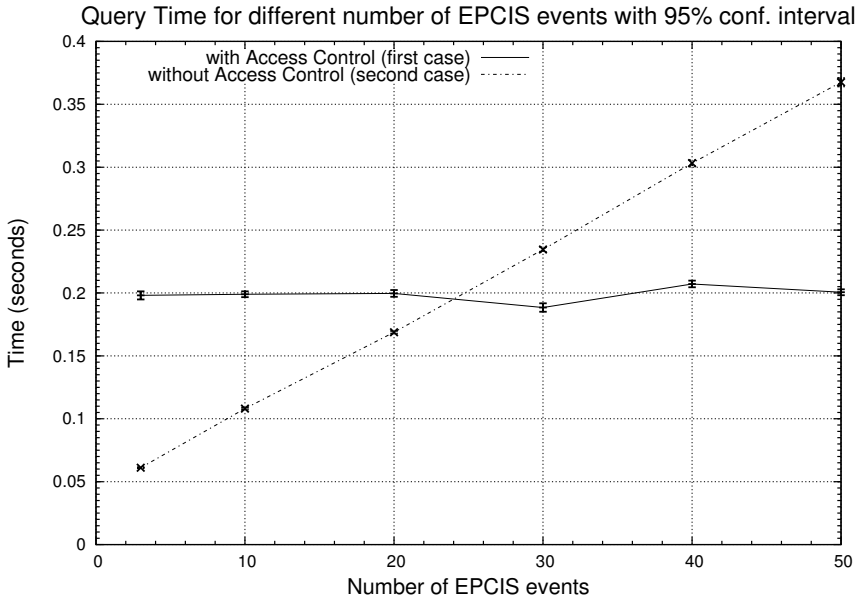

Figure 2.4: Query delay time for different number EPCIS events in the repository with 95% confidence interval

The results of the second scenario depicted in Fig. 2.4 shows that the number of EPCIS events data queried from the repository does impact the query time delay linearly. Obviously, the query delay time of the *first case* is relatively constant since it always return one EPCIS event as a result of the access control mechanism. Based on this finding, it can be concluded that both cases would achieve the same query delay time when the ratio of the EPCIS events data returned between the *first* and *second* case is around

1 : 25. Consequently, the delay gap between both cases as shown in Fig. 2.3 would be bigger if the result of access control evaluation would return more than one EPCIS events data. This ratio could be improved through a more efficient implementation.

2.7 Discussions

Earlier in this chapter, we have mentioned several requirements of access control model in an inter-enterprise RFID system, and how the proposed model fulfilled those requirements. In this section, we will present several important features of a secure system – in particular an access control for big data system – that are constituted in the proposed access control model. Furthermore, some qualitative comparisons with other access control models will also be given.

- **High-granularity:** The proposed access control model provides high-granularity since it does not allows the access only based on *permit* or *deny* action, but based on some specific types of data or context information. The level of granularity can be defined in the access policy by the system or security administrator of the system.
- **High privacy:** The high-granularity of access control policy provides high level of privacy to the business activities related information stored in the EPCIS repository.
- **Trust:** The access policy is applied to each user automatically based on some pre-defined user profiles, and a user is assigned to a particular user-profile based on trust relationship and contextual information.
- **Flexibility:** The proposed access control policy model offers high flexibility which is favourable in a big data system. It allows relative time definition instead of fixed time, EPC ID definition based on the EPC pattern, and automatic user assignment to user-profile through trust information.
- **Inter-operability:** The proposed access control model complies with the EPCIS Specification which is an open standard. The proposed system is also implemented as a web service which is highly inter-operable, i.e. independent of any specific programming language or server implementation.

In comparison with other existing access control, the proposed access control model is better as compared to them in the following aspects:

- *Trust-based X-GTRBAC [4]*: Our proposed access control model allows a way to incorporate trust in assigning user-profile to users which is quite similar to the approach presented in [4]. However, [4] does not fit the requirement of providing high-granularity access to data, particularly in an inter-enterprise RFID system.
- *AAL [6]*: A quite similar approach of a fine grained access enforcement specially designed for the EPCIS events data that is proposed in our access control model, was also introduced in [6]. However, automatic way of assigning some access policy to a user was not considered in [6], which gives a lot of burden to the system administrator, thus unrealistic for the real system. Moreover, its SQL query rewriting method for the access enforcement, does not inter-operable with the EPCIS implementation that is not based on the Relational Data Base Management System (RDBMS).

2.8 Conclusion

Access control management in an inter-enterprise RFID system is a great challenge, since such system allows the tracking and monitoring of large numbers of things, i.e. a path towards realizing the IoT vision. The most challenging access control problems in such a system are in providing high-granularity access of RFID events data, known as EPCIS events, with flexibility and efficient access policy management over a very dynamic and huge amounts of EPCIS events data. A novel access control model to address these problems has been proposed in this chapter along with complete definition of access control model and evaluation through the system implementation of the model. The findings in the evaluation show that the proposed access control model is consistent and could achieve less query time delay than the inter-enterprise RFID system without access control if the ratio of the returned EPCIS events data is less than $1 : 25$.

The proposed access control model relies on the trusted third parties entity to obtain the user profile, but the mechanism on dealing with such trust management has not been addressed yet. Incorporating trust into the access control model, i.e. through PKI Certificate Authority (CA), is one possibility of the future work. Another direction targeted in the near future is to

improve the current system level implementation, especially in addressing the limitation described in Section 2.5.

2.9 References

[1] Bayu Anggorojati, Parikshit N. Mahalle, Neeli R. Prasad, and Ramjee Prasad. Secure access control and authority delegation based on capability and context awareness for federated iot. In Fabrice Theoleyre and Ai-Chun Pang, editors, *Internet of Things and M2M Communications*. River Publisher, 2013.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787 – 2805, 2010.

[3] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. Trbac: A temporal role-based access control model. *ACM Trans. Inf. Syst. Secur.*, 4(3):191–233, August 2001.

[4] R. Bhatti, E. Bertino, and A. Ghafoor. A trust-based context-aware access control model for web-services. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 184 – 191, july 2004.

[5] Rafae Bhatti, Arif Ghafoor, Elisa Bertino, and James B. D. Joshi. X-gtrbac: an xml-based policy specification framework and architecture for enterprise-wide access control. *ACM Trans. Inf. Syst. Secur.*, 8(2):187–227, May 2005.

[6] Eberhard Grummt and Markus Müller. Fine-grained access control for epc information services. In *Proceedings of the 1st International Conference on The Internet of Things*, IOT'08, pages 35–49, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] http://wiki.aspire.ow2.org.

[8] EPCglobal Inc. Epc information services (epcis) version 1.0.1 specification. September 2007.

[9] EPCglobal Inc. Gs1 epc tag data standard 1.6 - ratified standard. September 2011.

[10] J.B.D. Joshi, E. Bertino, U. Latif, and A. Ghafoor. A generalized temporal role-based access control model. *Knowledge and Data Engineering, IEEE Transactions on*, 17(1):4 – 23, jan. 2005.

[11] Tom Karygiannis, Bernard Eydt, Greg Barber, Lynn Bunn, and Ted Phillips. Guidelines for securing radio frequency identification (rfid) systems - recommendations of the national institute of standards and technology. *NIST Special Publication*, April 2007.

[12] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *Computer*, 29(2):38 –47, feb 1996.

[13] XACML. https://www.oasis-open.org/standards#xacmlv2.0.

3

Secure Open M2M-RFID Middleware

The addressing and networking are amongst the most important issues in order to make the IoT concept a reality. In the RFID based IoT context, the most challenging issue is on the integration of the RFID network into the IP network. The development of the RFID technology, which is driven mainly by the logistic related industries, is based on enterprise system that does not take full advantage of the IP based networking features, such as the mobility management. On the other hand, most of the existing works towards the integration of RFID and IP network do not take the full advantage of all the RFID middleware components defined in the EPCglobal standard. Therefore, the full integration of RFID middleware with the IP network through a testbed implementation which is fully compliance with the EPCglobal standard and having the location and mobility management at the same time, is the main contribution of this work. The performance of our proposed work is measured in terms of delay in two cases, i.e. tags registration and tracking, and thereafter it is compared with the existing RFID middleware implementation. The measurement results show that the delay of our proposed work is significantly lower than that of existing solution, especially in the tags registration case.

3.1 Introduction

The IoT is a novel paradigm in which the basic idea is to enable the interaction and cooperation of pervasive *things* or *objects*, such as RFID tags, sensors, actuators, etc, with their neighbors to reach common goals [2], and to be accessible through internet. RFID as one of technology that enables object level service on the internet, follows a certain standard, namely EPCGlobal standard [8], in order to realize the IoT vision on RFID.

One of the major problem in current RFID network architecture is that of its infrastructure has a limited accessibility in the global IP network. EPCglobal standard has been driven mainly by the industry sectors that realy utilize RFID and gain benefits from it, e.g. logistics and supply chain, in a form of RFID middleware that aims to inter-connect the RFID system with enterprises IT infrastructure. As a result, the existing EPCglobal specification is very enterprise and middleware "oriented", e.g. focuses on adopting web-service/SOAP technology which is computationally heavy for low-powered mobile device. In the context of the RFID tag tracking service, the EPCglobal RFID middleware relies on an entity called ONS which requires the client to access directly a computationally heavy service provided by the EPCIS query interface. Moreover, the root ONS which is an application of DNS has no support on mobility. By far, the mobility is not a very important issue since biggest customers of RFID technology are mostly logistic industries which do not really require real-time tracking. However, when a sensing capability is included within the RFID tag and real-time monitoring is required, mobility becomes an important part to be considered. By having all those aspects together in mind, the main objectives of this work are to increase the accessibility of RFID data over the internet with mobility support in the current EPCglobal specification, while keeping all the advantages of the EPCglobal based RFID middleware.

With its support in mobility management, SIP is a strong candidate to provide such functionality in the RFID. Besides that, SIP is widely used in many of the popular IP networking services such as Voice over IP (VoIP), tele/video-conference, IP Multimedia Subsystem (IMS), etc, thus suitable for the purpose of integrating RFID with the global IP network. However, most of the SIP based proposed solution did not leverage the rich functionality of RFID middleware as defined by EPCglobal standard. The main contribution of the proposed solution is that it takes the full benefit of the RFID middleware defined in the EPCglobal architecture, which allows us to obtain richer information such as business context information. Moreover, it provides the

RFID location management and integration with the Next Generation Network (NGN) through SIP functionality.

Besides the location management, security and privacy are of the most importance for the RFID system, especially when the RFID events data is accessible across several different organizations. In Chapter 2, a fine-grained and efficient access control for the inter-enterprise RFID system has been proposed. The proposed access control system provides solution to problems in such a system, such as granularity, privacy, trust, flexibility, and inter-operability. From the system architecture perspective, applying access control before the EPCIS component (i.e. at the Tracking module or the proposed SIP gateway) is intuitively considered to be more secure than at the EPCIS because it acts as the first "gateway" in providing access to the EPC tags ID tracking request before reaching the EPCIS. However, it is necessary to evaluate the performance of the proposed location management architecture for the RFID system combined with the access control model proposed in Chapter 2 to measure the effectiveness of such methods objectively.

The remaining of this chapter will be organized as follow. The architecture EPCGlobal and SIP network is presented in section 3.2. Section 3.3 describes our proposed solution in terms of architecture and protocol.The testbed implementation of our proposed architecture is explained in 3.4. Performance evaluation based on the testbed implementation along with some discussion is presented in section 3.5. Finally conclusion and future insight are given in section 3.6.

3.2 Related Works

An approach to fully convert EPCglobal into SIP network to be compatible with IP based solution was proposed in [3]. It introduced an entity named Surrogate User Agent (SUA) which acts as a SIP Client, i.e. performing the SIP signalling that supports the location management. The types of SIP signalling message performed by SUA are *REGISTER* and *INVITE*. The *REGISTER* message is used in the location registration procedure and *INVITE* message is used in the tag tracking procedure. The authors of [3] also introduced an entity called SRMS Name Server (SNS) which has the similar role as DNS in the internet world or ONS in the RFID network, but it translates the EPC tag ID into a SIP URI format instead of translating URL into IP address or EPC tag ID into an RFID tracking or EPCIS query service interface. However, it was not clear how such translation is technically done. Moreover, with

regards to an RFID middleware functionality, SUA only performs tag filtering which is a small portion compare to the overall RFID capabilities according to the EPCglobal standard.

Another approach to integrate the RFID network with the global IP network was introduced in [12], in which an IP based infrastructure to retrieve EPC tag ID's location was proposed. The proposed architecture consists of *RFID readers*, *location database*, RFID agent, *location server*, and *name server*. In the proposed architecture, the *registration* and *tracking* are also introduced in order to manage the location of the RFID tag, and then a SIP-based RFID location management system is again used as a proof of concept. However, the paper does not explain clearly how the tag ID is translated into a name that can be recongnized in SIP, i.e. SIP URI, in fact the paper assumes a generic name such as *doctor1*. More importantly, the RFID middleware is not even utilized in the proposed architecture, thus the full advantage of RFID middleware cannot be leveraged by the platform.

ROADS (RFID Office Application for Document tracking over SIP) [11] is another attempt to integrate RFID with the global IP network. The paper emphasized its solution on the compatibility with NGN, in particular with the IMS which is also based on SIP. The SNS entity which was introduced in the [3], and SIP for Instant Messaging and Presence Leveraging Extension (SIMPLE) protocol which is a transport of Instant messages in SIP that works based on publish-subscribe principle are also introduced in the ROADS system [11]. Similar to the previous solutions, this system does not leverage the full advantage of an RFID middleware.

On the other hand, the development of RFID middleware based on the EPCgloblal standard has been done by several companies as licensed and proprietary solutions as well as by the open source communities. One of the existing open source RFID middleware is AspireRFID [10] which has been previously described in section 1.3.2. It introduced some components that extends the original EPCglobal standard, specially Tracking Service that provides the tracking service and interface between ONS and EPCIS. However, it relies fully on the web service implementation which does not support mobility management and it is heavy weight to be implemented in the mobile devices, such as smart phone and tablet.

3.2.1 Contributions to the existing works

According to the review of the existing works on the IP based RFID location and mobility management solutions as well as the RFID middleware, it

is clear that an integrated solution that combine both solutions into a single platform. The major contribution in this work is to realize an integration of an EPCglobal RFID with the global IP network based on SIP, which provides location and mobility management functionality and also leverages the EPC-global based RFID middleware capabilities. This means that the platform will be incorporated with the business contexts information and more light weight which means that the performance will be more efficient. On top of it, the platform is also integrated with the security access control designed especially for the inter-enterprise RFID systtem as explained in Chapter 2.

3.3 Proposed architecture

To achieve the objectives as described earlier in Section 3.1, a system architecture is proposed as depicted in Fig. 3.1. The proposed architecture consists of the components from AspireRFID middleware [10] and SIP components, along with the interfaces enabling their integration. Besides that, a protocol to allow communication between different components and identity translation between two different systems, e.g. RFID and SIP, is also proposed.

3.3.1 EPCglobal RFID Middleware – SIP Architecture

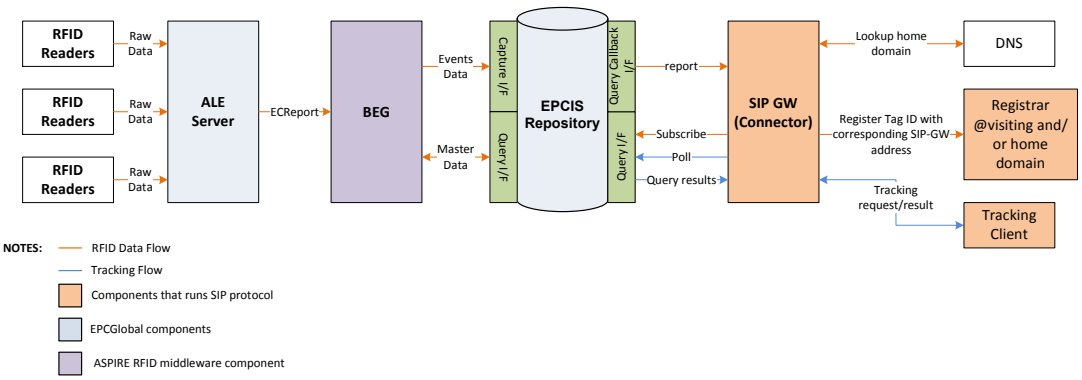

Figure 3.1: Proposed EPCglobal-SIP architecture

Each of the component in the proposed EPCglobal-SIP architecture shown in Fig. 3.1 is briefly explained in the following subsections in order to give an idea of how the overall system works.

3.3.1.1 ALE Server

It handles the connection with RFID readers and the collection of RFID raw data. Furthermore, it performs filtering of RFID tags with certain pattern of EPC ID and with certain kind of action, e.g. addition, current, or deletion of RFID tags passing the reader. Therefore, only RFID data that matches criteria sets by the filter can be passed to the next component of RFID middleware, e.g. BEG, in the form of ECReport (Event Cycle Report).

3.3.1.2 BEG

BEG is a component proposed in ASPIRE RFID Middleware [10] as an extension to EPCglobal architecture. It essentially processes the report from ALE server, and then send the data into the EPCIS Repository through the EPCIS Capture Interface. The forwarded data to EPCIS repository depends upon the business transaction that is defined in the BEG, known as Master Data. BEG is able to retrieve those available Master Data in the EPCIS repository through EPCIS Query Interface.

3.3.1.3 EPCIS

EPCIS is the heart of RFID middleware based on the EPCglobal architecture framework. Its main functionality is to process two kinds of data, namely Event Data and Master Data [6]. Given that, EPCIS transforms the Filtered raw RFID data into more meaningful information in regards to business transaction, e.g. the transaction ID, type of business step, business location, disposition, etc. In order to enable the sharing of information in the repository to other components, EPCIS has two types of interface called Capture and Query interfaces. In the query interface, there are two available methods to query the data in EPCIS repository, namely "poll" and "subscribe". The "poll" method allows the query results to be returned once it is invoked, while the "subscribe" method allows the subscriber to retrieve the query results periodically according to the time period that is defined in the subscription. Moreover, the "subscribe" method requires a so called Query Callback interface to allow it sending a periodical query report to the subscriber.

3.3.1.4 SIP Gateway

SIP Gateway is the main component that enables the communication between RFID middleware and other devices through SIP protocol. In the proposed architecture, SIP gateway has two main functionalities: location or mobility management of RFID tags using SIP protocol, and tracking of RFID tag. Based on the architecture depicted in Fig. 3.1, those two main functionalities are represented by orange and blue lines. The mobility management which is represented by orange line is accomplished by applying "subscribe" method to the EPCIS Query interface which returns periodical query report, and then by invoking a SIP "REGISTER" message any time a new tag ID is reported. On the other hand, the tracking of RFID tag is initialized by a SIP "MESSAGE" sent by a tracking client that contains a tracking request. Upon receiving a tracking request message, SIP gateway invokes a "poll" method towards EPCIS which returns some results and finally sends the tracking results back to tracking client through another SIP "MESSAGE". It is important to mention that, the benefit of having a SIP based RFID tag tracking is to allow the tracking client to access the EPCIS repository through the SIP gateway which is more light-weight than the direct access to the EPCIS query interface, that is done in the traditional ONS based tag tracking.

3.3.2 SIP Protocol

The proposed architecture as explained earlier will not be able to perform mobility management and tracking of RFID tags functionalities properly without the support of SIP protocols as well as its infrastructure. With that in mind, the details of SIP protocols along with the infrastructure supporting the proposed architecture will be explained in this section.

3.3.2.1 EPC ID-to-SIP URI translation and address resolution

In order for an RFID tag with a certain ID that follows EPCglobal standard to be "connected" in the internet world through SIP protocol, there are at least 2 steps required: EPC-to-SIP URI translation and address resolution itself. The EPCglobal has defined ONS standard [7] that allows an EPC tag ID to be resolved as an internet address through a specific DNS implementation. However, it does not support the EPC ID -to-SIP URI translation and further resolve it into an IP address along with the communication port and transport protocol. [3] introduced a new entity called SNS (SIP Naming Service) that

translates EPC-ID into SIP-URI and defines how the address translation is done hierarchically. However, [3] does not explain clearly how the SIP address resolution is being carried out. Given with limited information in the literature, we proposed a method that consists of 2 steps as mentioned earlier.

An important assumption in the translation process is that the EPC ID translation from the initial reading of RFID tag in the form of pure Hexadecimal or Binary format into a URI format has been done by a TDT [9] module, which practically can be part of the ALE server or running as a standalone service that is connected to ALE server. Given this assumption, a method to translate an EPC ID in a pure identity URI form into a SIP URI is proposed.

A typical EPC ID in a pure identity URI form would be following a general syntax as follows:

urn:epc:id:EpcScheme:CompanyPrefix.ItemReference.(SerialNumber)

The EpcScheme can be gid, sgtin, grai, etc. The serial number is an optional field, since it is applicable in some of EpcSchemes but inapplicable in others. For example, the following EPC ID in a pure identity URI format: *urn:epc:id:gid:6776251.255.480*, is mapped into the following SIP URI: *sip:480.255@6776251.gid.id.anydomain.com*. This means that the SIP domain of the URI example *6776251.gid.id.anydomain.com* and the SIP user ID is *480.255*. Having the EPC ID mapped into the SIP URI, the complete address resolution of SIP URI along with the port and transport protocol is done based on the guideline in RFC 3263 [1]. Practically speaking, the address resolution relies on DNS infrastructure which is quite similar to the ONS, but its detail implementation involves configuration of the SeRVice (SRV) record in DNS besides the NAPTR record.

3.3.3 Access Control Module

The detail implementation of the access control module is thoroughly explained in Section 2.5. The main difference on how the access control is integrated in the overall architecture between the one proposed in Chapter 2 and this current chapter where a SIP gateway is introduced, is that the access control in Chapter 2 is performed at the EPCIS server level (as seen in Figure 2.2), while in this chapter it is performed at the SIP gateway level. Upon adding the access control module into the proposed EPCglobal-SIP architecture, the architecture will look like as it is depicted in Figure 3.2.

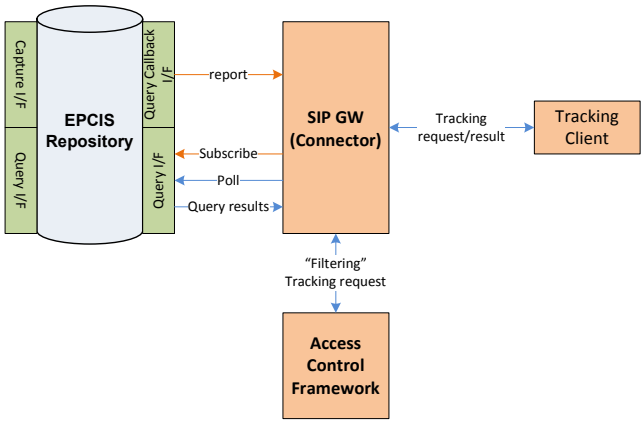

Figure 3.2: Proposed EPCglobal-SIP architecture

3.4 The Testbed implementation

The testbed consists of RFID middleware components, e.g. ALE server, BEG, EPCIS server, and Accada RFID reader simulator, that are developed and adapted from the ASPIRE project [4], SIP gateway to the EPCIS middleware component, tag tracking client based on SIP protocol, and two SIP proxy servers based on OpenSIPS project implementation [5].

The SIP gateway implements both the SIP protocol and EPCIS query interface as shown earlier in Fig. 3.1. Moreover, the SIP gateway is also capable of translating the EPC-ID into a SIP URI format and then resolving the SIP URI based on the guideline in [1] as explained previously in Section 3.3.

To explain better how the whole procedures work, a simple scenario will be presented. The scenario being considered in this work is first of all a retail company ***Ret*** receives some items with RFID tags from a producer ***Prod*** in its receiving RFID gate. When the items arrived at the RFID gate of ***Ret***, the middleware processes the tags and other information, and finally the tags registration into the SIP servers owned by ***Ret*** and ***Prod*** is carried out. On the other hand, a user who would like to know the latest location of the item and other information, would use a tracking application and receives information that he/she requests for. Two cases for the measurements are being done. The first one being the registration of tags through SIP protocol and the second one is the tags location tracking.

The overall procedure of tag registration process involving the query subscription to EPCIS server and registration using SIP protocol is presented in

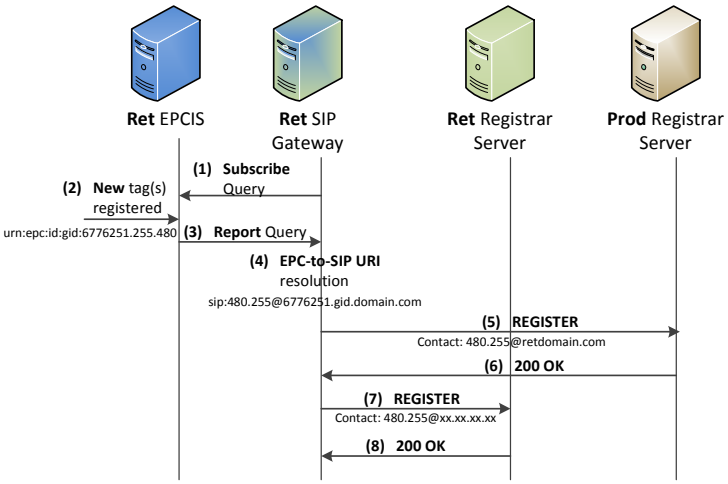

Figure 3.3: Tag registration procedure

Figure 3.3. The tag registration process relies on the subscribe method of the EPCIS query interface in which the SIP subscribes to a particular event in the EPCIS (step 1 in Figure 3.3), i.e. a new tag is added into the EPCIS repository (step 2 in Figure 3.3), and then a periodic message with a configurable time interval would be sent to the SIP gateway as long as the event matches the subscription criteria (step 3 in Figure 3.3). As soon as a new tag is read by the reader and processed by the middleware up until the SIP gateway, the tag registration process by using SIP protocol is performed. Before that, SIP gateway would perform an EPC-to-SIP URI translation as well as a SIP address resolution by using a DNS query (step 4 in Figure 3.3). Since the tag is received at the ***Ret*** premises which is considered as the foreign domain to the ***Prod***, the tag registration is done to both ***Ret***'s and ***Prod***'s Registrar Servers. Upon the registration at the ***Prod***'s home registrar server (step 5 in Figure 3.3), the domain part of the *Contact* address in the SIP header contains the ***Ret***'s domain, which in this case is essentially corresponds to the ***Ret***'s Proxy Server. The purpose of this is that when a tracking request is received at the ***Prod***'s Proxy Server, the request will be forwarded to the ***Ret***'s Proxy Server which will be explained in a more detailed when we describe the tracking procedure later on in this section. Further on, upon the the registration at the ***Ret***'s local Registrar Server (step 7 in Figure 3.3), the domain part of the *Contact* address in the SIP header contains the IP address of the SIP gateway which is the main contact point in querying the requested EPC tag ID to the EPCIS server. Finally the 200 OK message will be sent back to the SIP gateway upon successful registrations (step 6 and 8 in Figure 3.3).

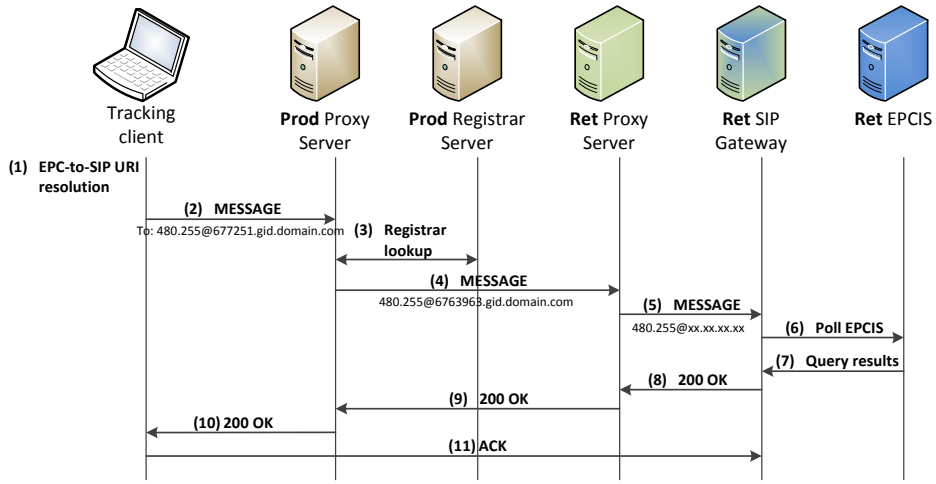

Figure 3.4: Tag tracking procedure

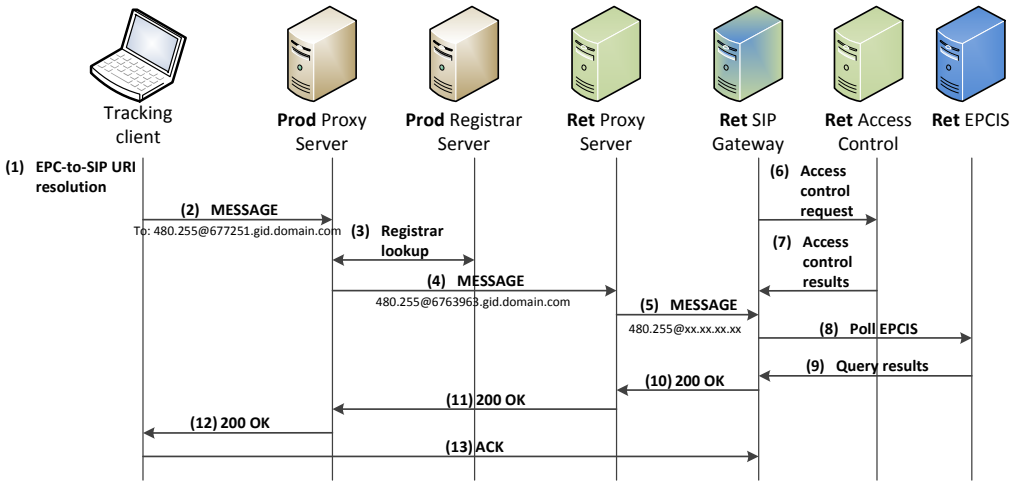

Figure 3.5: Tag tracking procedure with access control

The overall procedure of tag tracking process is depicted in Figure 3.4. First of all, the tracking client would translate the EPC tag ID to be tracked into its corresponding SIP URI (step 1 in Figure 3.4). After the SIP URI is obtained, a SIP message that contains some query parameters, such as EPC ID, date, and time, is sent to the ***Prod***'s Proxy Server (step 2 in Figure 3.4). In the implementation, the query parameters are formatted in Java Script Object Notation (JSON) format. The ***Prod***'s Proxy Server would do a lookup of the given SIP URI stated in the *To* field of the message's header, in its Registrar (step 3 in Figure 3.4). After the *Contact* address, which is the ***Ret***'s domain, is found in the Registrar, the ***Prod***'s Proxy Server would resolve its internet address and then forward it (step 4 in Figure 3.4). Upon receiving

the message, ***Ret***'s Proxy Server would forward it to the SIP gateway (step 5 in Figure 3.4), that can be obtained from the ***Ret***'s Registrar after the registration process. Further on, the SIP gateway would invoke a poll method through the EPCIS query interface, unlike in the tags registration process that uses subscribe method (step 6 in Figure 3.4). The EPCIS repository would then return the query results to SIP gateway according to the query parameters that are requested (step 7 in Figure 3.4). Upon receiving the query results, the SIP gateway would send a 200 OK response message including the query results down until the tracking client side (step 8-10 in Figure 3.4). Finally, an ACK message is sent to the SIP gateway (step 11 in Figure 3.4). Upon integrating the access control mechanism in the tag tracking procedure, some additional steps are required in the overall tracking process as shown in Figure 3.5. Those additional steps are essentially an access control request and a set of access control resutls sent by the SIP gateway to the access control module and vice versa (steps 6 and 7 in Figure 3.5).

3.5 Results and discussion

3.5.1 Measurement procedures

A series of experiments have been carried out to measure the performance of the proposed system architecture in both tags registration and tracking cases, which is particularly interesting due to the fact that a high number of tags are being read at one cycle in logistics, shipping, and such. The parameter that is being measured is the time consumed to perform each operation over different number of tags. The time consumed for the tags registration is defined as the time between the first SIP REGISTER request message among all the tags being sent and the last SIP 200 OK response message being received by the SIP gateway. Similarly, the time consumed for the tags tracking is defined as the time between the first SIP MESSAGE being sent and the last received SIP 200 OK response message. For each number of tags, 700 measurements have been done for the tags tracking case, and more than 600 measurements were taken for the tags registration case. The measurement results of our proposed method are then compared with the ONS Tracking Service from ASPIRE [4], which will be referred as the reference method, under the similar scenario. It is important to note that there exists other traffics other than the traffic for the measurement purpose, e.g. LAN and internet traffic, during the experiment due to the needs of the DNS. As a result, that factor greatly affects the delay of

some tiny numbers of measured data which were discarded in the final statistics analysis.

3.5.2 Analysis of measurement results

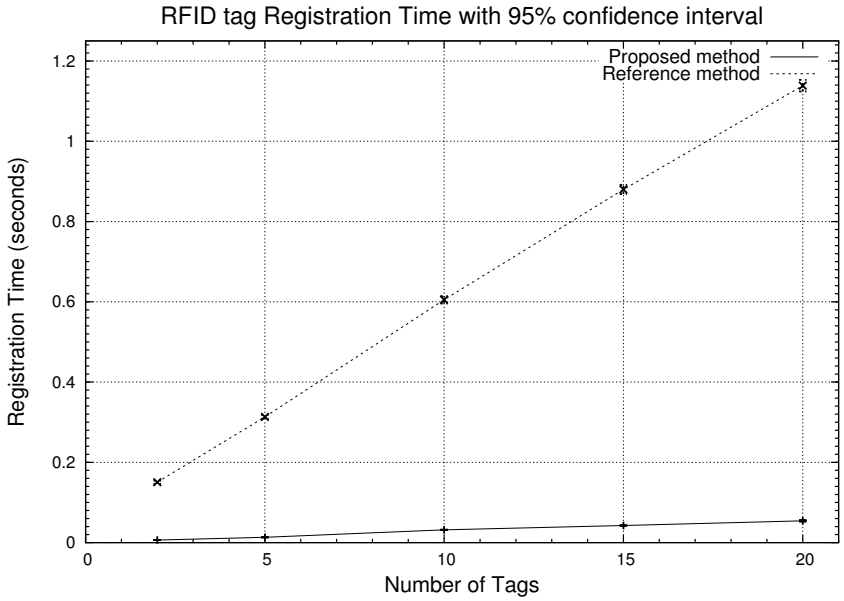

Figure 3.6: Time consumed in the registration process for different number of tags with 95% confidence interval

The measured delay as a result of the experiments for tags register and tracking cases are depicted in Fig. 3.6 and Fig. 3.7 respectively. Both measurement results show clearly that the delay in tag registration and tracking mechanisms of the proposed method are lower then the one being compared, especially in the registration mechanism in which the delays differ significantly. The main reason behind it is that because the reference method fully utilizes web based operation that takes relatively high delay to perform, such as the tracking service methods as well as the EPCIS poll query, which involves web service methods invocation and database query. That is why the delay of reference method is significantly higher than the proposed method which is fully utilizes SIP protocol in the tags registration case. This argument is even more confirmed when the two cases of our proposed method are being compared. Here, we can obviously see that the tags tracking process takes roughly more than 10 times the delay of the other case. This is due to extra signalling and processing within the SIP Proxy servers, e.g. registrar lookup and address

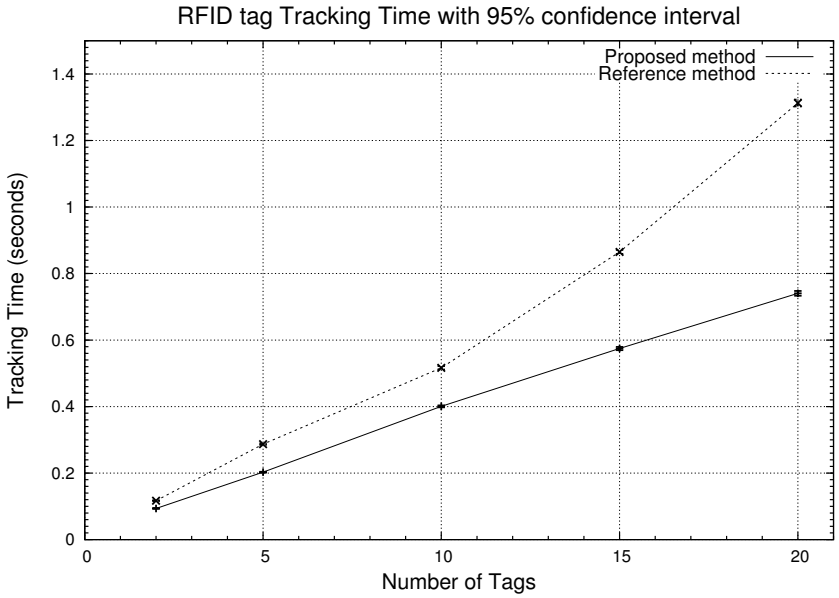

Figure 3.7: Time consumed in the tag tracking for different number of tags with 95% confidence interval

resolution, as well as the query operation to the EPCIS server, that it has to perform. In fact, the query of EPCIS server takes around 5/6 of the total delay in the tags tracking process. Even so it is still higher than the delay of tags registration process which is obvious since more SIP signalling is involved, the process involving SIP signalling alone in the tags tracking procedure is significantly lower than the delay of the web based operation. Based on the findings in Figure 3.7, we can also conclude that the proposed method is more scalable than the reference one under the same condition. This behaviour is due to the database query in the reference method is done twice, that is two times higher than the proposed one, which makes the delay difference between two cases increases almost double as the number of tags increases.

Furthermore, confidence interval has been calculated in order to check the reliability of the data. As stated earlier, a series of experiments of more than 600 measurements for each scenario which is independent to each other and results in a large enough population of data, implies that the distribution of the sample mean approach a normal distribution. This is true according to the central limit theorem for a large enough sample size as long as the population has a finite standard deviation. Therefore, it is safe to assume that the sample data has normal distribution which is then used in deriving the 95% confidence interval values of the measured data, which are presented in Table 3.1.

Table 3.1: 95% confidence interval values of the measured data

	Proposed		Reference	
	Register	Tracking	Register	Tracking
2 tags	0.000134	0.000471	0.004006	0.001357
5 tags	0.000208	0.001112	0.004607	0.002315
10 tags	0.000618	0.002819	0.009175	0.005213
15 tags	0.000672	0.004948	0.011330	0.005799
20 tags	0.001979	0.007190	0.014357	0.010330

The confidence interval general trends show that, the higher the number of tags the higher the 95% confidence interval values, although the values over different tags does not have linear relationship. As the confidence interval value is highly influenced by the standard deviation, this finding shows that the measured delay values varied a lot especially when high number of tags are used. One of the reason is that because the message in the either case is sent as an individual request message that represents an individual tag ID, especially for the case of the proposed method and the tags tracking case of the reference method. Therefore, the probability of the message being delayed, e.g. due to packet retransmission or congestion in the network, is higher when more individual messages are sent as a whole.

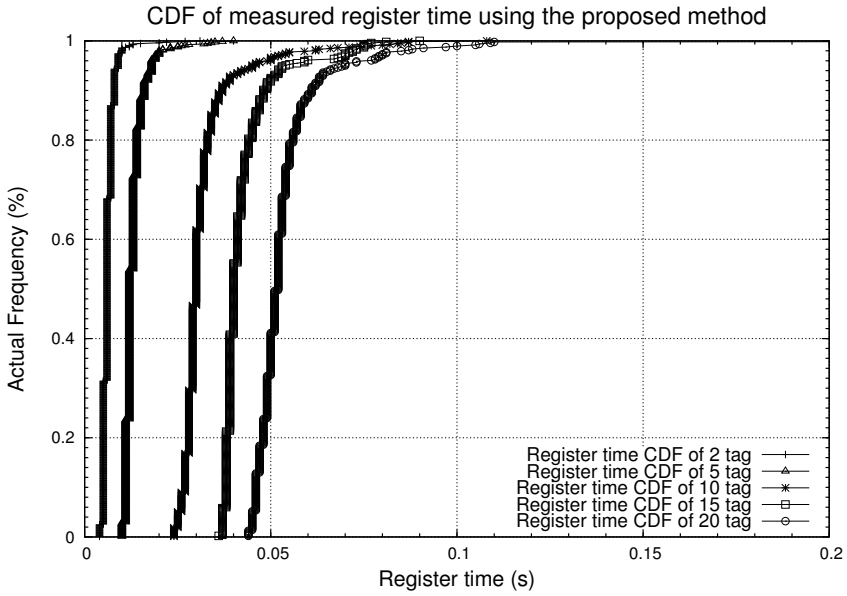

Figure 3.8: CDF plot of the proposed method in the registration processes for various tags numbers

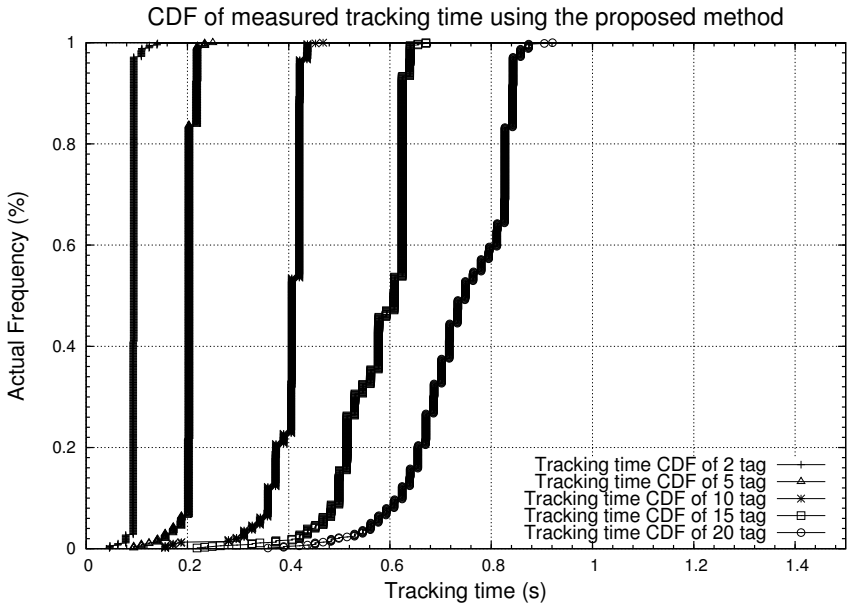

Figure 3.9: CDF plot of the proposed method in the tracking processes for various tags numbers

In addition, the Cummulative Distribution Function (CDF) plots of the proposed method for both registration and tracking cases are shown in Fig. 3.8 and 3.9. Those CDF plots confirm the previous results that the SIP tag registration process takes very low delay, i.e. less than 60 milliseconds for 20 tags in average, and the confidence intervals increase as the number of tags is increased in both cases.

3.5.3 The impact of access control in the tags tracking case

In order to test the impact of access control in the tags tracking case in the proposed system architecture and the reference system, another series of measurement is performed similar to the procedure carried out in 3.5.1 for the tags tracking case. The measured delay of the experiment for tags tracking case with the impact of access control is presented in Figure 3.10. The measurement result shows a peculiar behavior of the proposed system architecture when the access control is applied, especially when there is 15 or more number of tags to be tracked. Initially, the tags tracking time of the proposed system architecture combined with the access control mechanism is comparable with the reference system even without access control, i.e. in the range of 0.5 second,

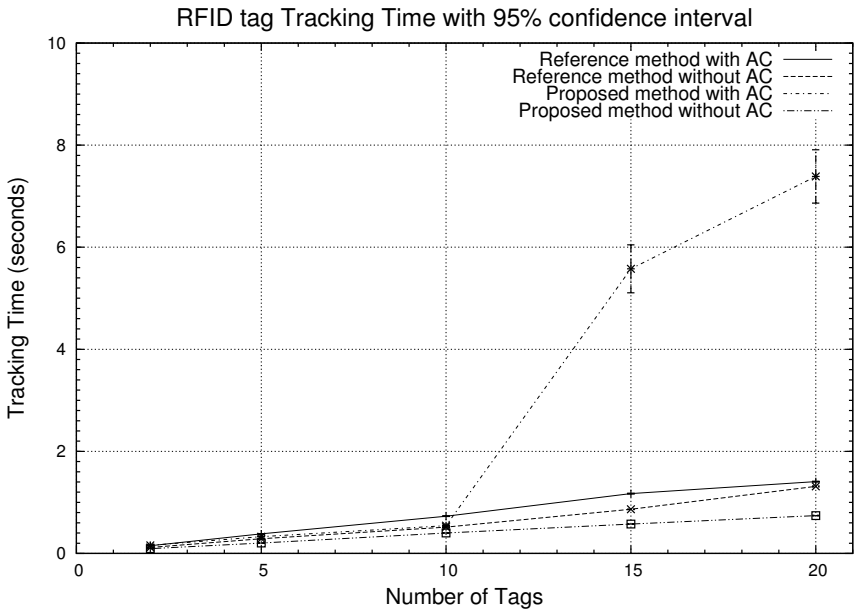

Figure 3.10: The impact of access control on the time consumed in the tags tracking for different number of tags with 95% confidence interval

for 10 or less number of tags. But it goes up as high as 5.5 seconds and 7.4 seconds when the number of tags is 15 and 20.

The main reason of such strange behavior is because the access control framework is implemented in web service, which is similar to the reference system, while the proposed system is based on SIP. As previously shown in Figure 3.5, the overall tag tracking procedure which is based on SIP protocol is "interrupted" by a time consuming web service based access control mechanism. In addition, the proposed tracking procedure requires the multiple tags to be tracked one by one. As a result, the protocol necessitates the invoking of access control service in multiple times, depending on the number of tag to be tracked, thus the tracking time grows extremely high when the tags reach a certain number, e.g. 15 tags or more according to the results depicted in Figure 3.10. This problem does not occur in the reference system because both the reference tracking method and the access control rely on web service. Furthermore, the web service operation is invoked only once regardless of the tag numbers, i.e. one web service invocation each for the tracking and access control. In summary, this result tells us that the access control mechanism needs to be implemented in a more efficient way, yet maintaining the interoperability, i.e. compatible with the overall RFID middleware as well as the specific location management implementation method.

3.6 Conclusion

The main contribution of this work is the integration of the EPCglobal based RFID architecture with SIP to their full advantage, i.e. providing functionalities of RFID middleware as well as RFID mobility and location management. A testbed implementation has been developed and a series of experiments have been carried out to measure the performance of the proposed architecture in terms of delay, for tags registration and tracking scenarios. Moreover, the measurement results of the proposed architecture are then compared with a reference method from an existing RFID middleware implementation. The results show that the proposed method performs significantly better than the reference method in terms of delay and scalability.

Some future directions of the development in this work include the integration of the system with IMS, integration with cloud platform either at the RFID middleware component or the SIP as well as IMS network entities, and an implementation of the security and privacy mechanisms.

3.7 References

[1] Rfc 3263 - session initiation protocol (sip): Locating sip servers. June 2002.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787 – 2805, 2010.

[3] Kideok Cho, Sangheon Pack, Taekyoung Kwon, and Yanghee Choi. Srms: Sip-based rfid management system. In *Pervasive Services, IEEE International Conference on*, july 2007.

[4] http://wiki.aspire.ow2.org.

[5] http://www.opensips.org.

[6] EPCglobal Inc. Epc information services (epcis) version 1.0.1 specification. September 2007.

[7] EPCglobal Inc. Epcglobal object name service (ons) version 1.0.1. May 2008.

[8] EPCglobal Inc. The epcglobal architecture framework. December 2010.

[9] EPCglobal Inc. Gs1 epcglobal tag data translation (tdt) 1.6 - ratified standard. October 2011.

[10] John Soldatos, Nikos Kefalakis, Nektarios Leontiadis, Nikolaos Konstantinou, Nathalie Mitton, Loc Schmidt, Roudy Dagher, Mathieu David, Bayu Anggorojati, Simone Frattasi, Neeli Prasad, Didier Donsez, Gabriel Pedraza, and Kiev Gama. D3.4b: Core aspire middleware infrastructure. Technical report, ASPIRE, 2010.

[11] P. Solic, N. Rozic, and N. Ukic. Roads: Rfid office application for document tracking over sip. In *Software, Telecommunications Computer Networks, 2009. SoftCOM 2009. 17th International Conference on*, september 2009.

[12] P.N. Tran and N. Boukhatem. Ip-based rfid architecture and location management. In *Software, Telecommunications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference on*, sept. 2008.

4

Secure Access Control and Authority Delegation based on Capability and Context Awareness for IoT

Access control is a critical functionality in IoT, and it is particularly promising to make access control secure, efficient and generic in a distributed environment. Another an important property of access control system in the IoT is flexibility which can be achieved by access or authority delegation. Delegation mechanisms in access control that have been studied until now have been intended mainly for a system that has no resource constraint, such as a web-based system, which is not very suitable for a highly pervasive system such as IoT. This chapter presents the CCAAC model including the authority delegation method, along with specification and protocol evaluation intended for federated M2M/IoT. By using the identity and capability-based access control approach together with the contextual information and secure federated IoT, this proposed model provides scalability, flexibility, and secure authority delegation for highly distributed system. The protocol evaluation results show that the capability creation and access mechanism of CCAAC is secure against a rigorous man-in-the-middle attack, e.g. eavesdropping and replay attacks, and is able to provide authentication as well.

4.1 Introduction

The Future Internet (FI) will be shaped by the rapid growth of the current internet in terms of penetration, services, capacity and contents along with ultra-small yet powerful mobile communication devices. The vision of internet connectivity by anyone, any time and anywhere, will be augmented by a new dimension, namely connectivity of anything driven by the advancements in the development of smart devices including WSN, RFID, Near Field Communication (NFC), etc. This is often referred to as IoT and enables different modes of communication, i.e. things-to-person, things-to-things and person-to-things communication along with various intelligent services and applications. A large number of research challenges remain in order to ensure the successful operation of IoT and some of the most significant of these relate to the capabilities of the network to offer security, privacy and trust.

IoT is ubiquitous and heterogeneous in nature. As a result, a large numbers of risks and threats emerge that can threaten the system's security and user's data privacy. Among others, access control is one of the main security issues to be addressed in IoT that is highly pervasive. A novel access control model – CCAAC is proposed to overcome this issue. Furthermore, as IoT is characterized by highly dynamic nodes connectivity and network topologies due to the ever-changing nature of wireless channel, mobility, and factors such as limited power, a dynamic and flexible design of access control system that is suitable for IoT is of the most importance. For this purpose, the proposed CCAAC model also considers the delegation of authority in order to gain access to a certain resource which belongs to different security domain in a Federated IoT network.

The chosen approach for the model with access control based on the capability concept, and in particular the Identity based Capability (ICAP) system [8], is considered in order to cope with the scalability of IoT system since it is well suited for providing authentication and access control in distributed systems. By using the capability concept, the proposed model to some extent also contributes to the authentication mechanism, which will be shown later in Section 4.7. Furthermore, in order to fulfil the flexibility and adaptability in IoT, context awareness and dynamic security as well as privacy policy enforcement is applied in the proposed model. The Context Aware Security Manager (CASM) model originally proposed for a PN [18] will be used as a reference point due its suitability in distributed systems, especially in the case of federated IoT. Along with that, flexible access control system by means of authority delegation method based on a dynamic capability propagation is also included in the proposed model [1].

4.2 Related Works on Access Control and Authority Delegation Models

Capability based access control was derived from the capability system concept in the computing world [15] in order to address the shortcomings of ACL based models. Both capability based access control and ACL are two complementary traditional access control systems derived from an Access Control Matrix (ACM). Basically, an ACM consists of several columns that list the objects or resources to be accessed, and rows that list a set of subjects or users who have privileges to access different resources. Each row in the ACM can also be seen as a capability owned by a subject that needs to be presented whenever it wants to access any object or resource in a system.

A simple example of capability used in the real life is a ticket that allows a passenger to embark on different modes of city transportation, e.g. bus, tram, train, etc. If the ticket contains all the transportation modes in which the passenger can embark, then the capability based access control is used. On contrary, if the system check whether the credential contained in the ticket is listed in the access control list, then the ACL is used. In comparison [3], capability based access control is better than ACL in tackling e.g. the confused deputy problem and other security threats, it is more robust due to its distributed architecture, it supports more levels of granularity, access right delegation and revocation. The Identity based Capability or ICAP was proposed in [8] which solves the problem of capability propagation and revocation in classical capability based access control.

In Context aWare Access Control (CWAC) [13], the surrounding context of subjects and/or objects is considered to provide access but it has scalability issues and no support for delegation and revocation. In Context Aware Role Based Access Control (CARBAC) [14], context is integrated with RBAC dynamically which is defined as characterization of the surrounding entities for performing appropriate actions. With CARBAC each different role can be associated with certain contexts. For instance, a Professor role in a university can be associated with some contexts, such as PhD degree, x years of research experience, and y numbers of journal publications in the related field. However, improper association of context and roles results in inefficiency in scalability and time and context dependency results in complicated delegation.

A Personal Trusted Device (PTD) [2] is used to provide access to locked resources and places, which consists of access controllers, the PTD and an administration point, while embedded security of the PTD is not considered. In [16], an access control model is designed as standalone components each

managing its context and thus limiting the reuse and context sharing. [5] and [9] proposed solutions for decoupling the applications from the context resources which require permanent access to the server. However, in a Wireless Personal Area Network (WPAN) as one scenario considered in PN, access to fixed infrastructure cannot be guaranteed due to constraints like frequently changing connectivity, resource unavailability and battery power. In [17], an identity establishment scheme for IoT as well as authentication mechanism based on ECC and capability was proposed.

Research on delegation of authority using capability has been investigated in [11] and [10]. [11] addressed the issue of role and/or permission delegation based on a RBAC model in a cross-domain environment using capabilities. The central idea behind their proposed mechanisms was the mapping of capabilities into roles and permissions in each domain. Hasebe et al. extends the approach presented in [11] by adding the delegation of task to be performed in the model for workflow systems in [10].

Research on dynamic authorization delegation in federated environment between entities or machine-to-machine delegation has been reported by Gomi et al. in [7]. Authors of the paper investigated chain of delegation in multiple entities and how to provide secure delegation framework. To better illustrate the delegation chain in multiple entities being investigated in [7], we can think of a user who is authorized in his company's web server and wanting to access some information from other company's server, i.e. destination server. In this scenario, his request has to go through his company's server and other intermediate servers in between before reaching the destination server. Consequently, the user has to delegate his authority over the chains of intermediate servers in a relay-like fashion. The delegation framework proposed in [7] also introduced Delegation Authority and Authentication Authority entities to enable such authorization delegation. Another related work in this topic has also been reported by the same author, Gomi, in [6]. Unlike his previous work in [7], [6] focuses on the user-to-user delegation and does not consider multiple entities delegation. The main contribution was a delegation framework in federated environment using an access token, regardless of the access control model being used. In this context, an access token is an opaque string representing a delegator's authorization to delegate their privilege to a delegatee whom they specify, which does not contain any information about delegator's credentials. The paper also explains the mechanism of issuing a token, asserting it into an authorization document, and service provisioning based on the delegate token.

The existing delegation methods are designed mainly to serve web-based services which involve a large IT infrastructure. These kinds of delegation models are not practically visible for an IoT system, which has a lot of constraints, e.g. power, memory, etc. It is important to mention that the delegation of authority by means of capability propagation is part of CCAAC overall design model. Therefore, delegation method in CCAAC is not an extension of any existing access control model, e.g. RBAC, as presented in most of the previous works.

The federated IoT networking has not been much discussed in the literature. There is a significant amount of work done in the context of a federated device-to-device or machine-to-machine communication known as PN federation, which includes the networking, management, security framework and identity management in [18]. Federated IoT environments based on the PN concept [18] will be further elaborated in Section 4.3.2.

4.2.1 Contributions to the existing works

According to the review of the related works on access control and authority delegation, some challenges are still left open, especially in the context of IoT. The biggest challenge on the access control for IoT are the scalability and flexibility, not to mention the secure access control. As we understand already that the IoT is highly pervasive and distributed in nature, thus scalability aspect of access control system is one important characteristic to have. While access control system is usually centralized or "distributed" in a particular entity, e.g. server, our approach is to distribute the complexity to the accessing entity by using capability. The flexibility challenge in access control here refers to two things: the flexibility in the process of access decision making by considering the contextual information and the flexibility in terms of authority delegation. Based on the review of challenges related to such requirements in Section 4.2, the proposed access control scheme will tackle both aspects at the same time.

4.3 System Architecture

4.3.1 System architecture to support the CCAAC model

The system architecture for supporting the CCAAC model is depicted in Figure 4.1. It is important to note that the Personal Network (PN) [18] has been referred in this work due to its advance networking concept for device-to-device

communication that opens a path as one candidate of network implementation in IoT. Correspondingly, the security framework brought up within the PN would be a good starting point in designing a security framework, considering their similarities in characteristics and requirements.

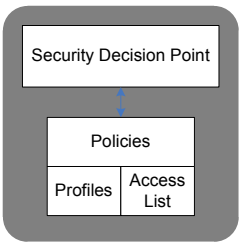

Figure 4.1: System architecture for supporting the CCAAC

Security Decision Point (SDP) in Figure 4.1 is the most important module in the system architecture to receive the capability creation or access request and make the final decision of the request. *Policies* serves as Policies Repository that consists of a collection of various policies for accessing available resources or objects. *Profiles* serves as Profile Repository, which essentially consists of subject as well as object profiles. Both of these components will be referred to as Policies Repository and Profiles Repository later on in Section 4.4. Finally, *Access List* plays an important role in supporting the capability-based authority delegation of access control by controlling the capability propagation as well as revocation through maintenance of a propagation tree as proposed in [8].

In principle, a new propagation tree is created at the *Access List* whenever a new capability is created, with the original capability owner becomes the root of the tree. When the capability owner delegates his authority by propagating his capability to someone else, then the propagation tree is updated by adding a new node. The one who propagated the capability can revoke the authority delegated to someone else before by requesting a revocation message to *Access List*. If the revocation message is valid, the corresponding node will be removed from the propagation tree.

4.3.2 Federated-IoT

Identity "Federation" is known term within the web security world and refers to management of a web user's identity across different security domains. The main reason of enabling federation in the web environment is that the work flow of the system often requires a user that is authenticated in one domain

to be authenticated in other domains as well. However, a concrete definition of the Federated IoT which is conceptually different as compared to the web security world, needs to be determined before addressing the issue of authority delegation in such environment.

First of all, the identity in the web-based system refers to a person's identity while in IoT, identity refers to a device or "thing". Therefore, the interaction of identities in IoT is in the form of device-to-device communication.

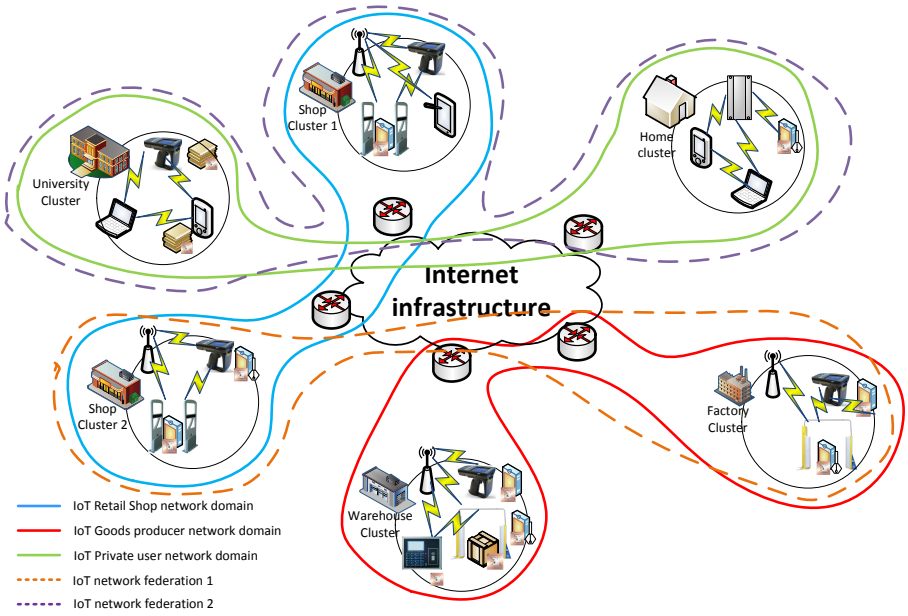

Figure 4.2: An example of Federated IoT network with delegation scenario

An example of federated IoT network in our context is depicted in Figure 4.2. Three IoT network domains are considered, i.e. private user, retail shop, and goods producer network, and two IoT-Federated networks are considered. An IoT network domain consists of one or more IoT cluster and the inter-cluster communication can be done either through the Internet infrastructure as shown in Figure 4.2 or through a wireless ad-hoc connection. Device-to-device communication within a cluster, i.e. intra-cluster communication, can be carried out by using different wireless access technology, e.g. RFID, ZigBee, bluetooth, wifi, etc.

4.3.2.1 Motivating scenario for access delegation

A good scenario example of how an authority delegation is needed in the Federated IoT networks can be shown as an inter-working of an intelligent-shopping service offered by a retail shop and a smart-fridge device that belongs to *private user* domain. The retail shop offers an intelligent service to its customer such that it is able to suggest a list of foods items based on their availability or expiration dates in the smart-fridge. In order for the retail shop to be able to come up with list of suggestions, it needs a privilege to access the smart-fridge which belongs to different network domain. To solve this issue, a valid privilege will be delegated to a device belongs to *retail shop* network. Before access delegation can be performed, it is assumed that the IoT network federation has been created, hence network authentication and trust relationship between two network domains have been established.

Another scenario of the authority delegation is when a device owned by a *private user* is trying to access the shop cluster 1 which belongs to *retail shop* network domain. In this scenario, suppose that a user that has a "smart fridge" at home is currently at the retail shop 1. He is checking the foods at home that need to be resupplied, at the same time checking those foods that may be available in the shop by accessing any wireless network device in retail shop 1. Since he does not belong to *retail shop* network domain, he needs a privilege in order to read information required by him. To solve such problem, a device that belongs to shop 1 cluster can delegate its privilege to private user's device. Before access delegation can be performed, it is assumed that the IoT network federation has been created, hence network authentication and trust relationship between two network domains have been established. The proposed delegation mechanism and the infrastructure to support the authority delegation will be presented in Section 4.6.

In such authority delegation scenario, some privacy problems could arise. For example, the *private user* do not want to reveal all food or drink items in his smart-fridge to the *retail shop*. With the authority delegation, the *private user* can delegate some of his access rights to *retail shop*. It should be noted that delegating authority does not mean to give full authority to other party as if the other party has the same role as the one delegating it. Furthermore, the delegated authority can be revoked at any time.

4.4 Proposed CCAAC Model

The proposed CCAAC model also includes properties of ICAP and context aware security, especially the concept of CASM and Virtual Identity (VID) [18]. This fills the gaps in current solutions as emphasized in Section 4.1.

4.4.1 Proposed capability structure in CCAAC

In the remaining of this chapter, the notations listed in Table 4.1 will be used.

Table 4.1: Notations used in the CCAAC definition

$\mathcal{S}$	Representing identifier of *Subject* that requests an access.
$\mathcal{O}$	Name of object or resource to be accessed.
$\mathcal{AR}$	Type of access right, e.g. read, write, execute.
$\mathcal{C}$	Context information.
Rnd	Random number generated from a one-way hash function to prevent forgery.
${}_{ext}CAP$	External capability that is held by the *Subject*.
${}_{in}CAP$	Internal capability that is kept in the *Object*, server, or whatever physical entity that is be accessed.
VID	Virtual Identity.
$\mathcal{ID}$	Unique identifier of the *Subject*.
$\mathcal{P}$	A set of *Subject*'s profiles.
$\mathcal{C}$	A set of contextual information that are used to define the VID and *Policy*.

Firstly, in order to support the context awareness in ICAP, an additional field is added in the ${}_{ext}CAP$ for the Subject S_i that contains the context information, $\mathcal{C}$, related to the capability. By including this, the external capability structure in CCAAC is defined as follow:

$$ {}_{ext}CAP_i = \{\mathcal{O}, \mathcal{AR}, \mathcal{C}, Rnd_i\} \tag{4.1} $$

where

$$ Rnd_i = f(\mathcal{S}, \mathcal{O}, \mathcal{AR}, Rnd_0) \tag{4.2} $$

$$ Rnd_0 = f(\mathcal{O}, \mathcal{AR}) \tag{4.3} $$

The internal capability ($_{in}CAP$) that creates a pair with the $_{ext}CAP$ which is stored in the object itself or an entity that has higher "authority" over the object (e.g. in hierarchical type of network), is defined as follows:

$$_{in}CAP = \{\mathcal{O}, Rnd_0\} \tag{4.4}$$

where Rnd_0 is defined exactly as in Equation 4.3.

4.4.2 Basic definitions

The important definitions used in CCAAC are presented in this section in which PN is the targeted platform.

4.4.2.1 Contexts

The *Contexts* $\mathcal{C}$, that are used to define the VID and *Policy* is essentially a set of contexts ($\mathcal{C}_{Set}$) with different types ($\mathcal{C}_{Type}$). The type of context can be a concrete property such as time or location, but also security related context such as authentication and trust level. In order to apply the context in the access control decision, each of the context types has to be evaluated with a certain constraint ($\mathcal{C}_{Const}$).

The overall context definition in CCAAC can be expressed with the following notation:

$$\begin{aligned} \mathcal{C}_{Type} &\in \{authLevel, trustLevel, time, location, \cdots\} && (4.5)\\ \mathcal{C}_{Set} &= \{\mathcal{C}_{Type(1)}, \mathcal{C}_{Type(2)}, \cdots, \mathcal{C}_{Type(n)}\} && (4.6)\\ \mathcal{C}_{Const} &:= \langle \mathcal{C}_{Type} \rangle \langle OP \rangle \langle VALUE \rangle && (4.7) \end{aligned}$$

where OP is a logical operator, i.e. $OP \in \{>, \geq, <, \leq, =, \neq\}$ and $VALUE$ is a specific value of $\mathcal{C}_{Type}$. Finally, we defined $\mathcal{C}$ as a set of context constraint $\mathcal{C}_{Const}$ as follows:

$$\mathcal{C} = \{\mathcal{C}_{Const(1)}, \mathcal{C}_{Const(2)}, \cdots, \mathcal{C}_{Const(n)}\} \tag{4.8}$$

4.4.2.2 Policy

A policy is associated with certain VID(s) that describes VID(s) preferences upon allowing other entities to access them. Please note that the entity or subject requesting access is described by its profile, e.g. subject's attributes, in the policy. A policy in the proposed CCAAC model holds an important role in access control decision as well as any process involving capability creation and delegation. Hence, it can simply be defined as a set of rules with parameters related to the user as follows:

$$Policy \in \{\mathcal{P}, \mathcal{C}, \mathcal{AR}\} \tag{4.9}$$

It is important to note that since a *Policy* is linked with a VID, and a VID itself is already linked with an *Object*, therefore the *Object* $\mathcal{O}$ notation is not included in the *Policy* statement. Furthermore, unlike the definition of the VID, the $\mathcal{P}$ and $\mathcal{C}$ that are included in the *Policy* statement are related to the *Subject* $\mathcal{S}$ who tries to access the *Object* $\mathcal{O}$.

4.4.2.3 VID

An entity, i.e. subject or object, may have more than one identity, namely one main identity and numbers as other alias identities; this is referred as VID. A VID consists of a user identifier and a set of disclosure policies where the same disclosure policy can apply to different VIDs. Therefore, the relationship between VIDs and disclosure policies is a many-to-many, which can be implemented using a pointer or hash map.

A VID is also attached with a particular context as well as profile information of the corresponding user [18] and it can be assumed that the profile information can be pre-defined as a set of default profiles or customized to a specific VID. The profile is assumed to have an one-to-one relationship with the VID and, for the context, to have a many-to-many relationship with the VID.

Based on these relationships and assumptions, the VID is defined as follow:

$$VID \in \{\mathcal{ID}, \mathcal{P}, \mathcal{C}, Policies\} \tag{4.10}$$

The $Profile$ $\mathcal{P}$ in VID may consist of *Objects*' $\mathcal{O}$ attributes and personal information. $\mathcal{C}$ refers to *Contexts* that can be a security context, such as trust

level, authentication level as well as other common contexts such as time and location (see Subsection 4.4.2.1). The $\mathcal{ID}$ is a unique identifier that can be acquired through cryptographic operations. The *Policies* is a set of *Policy* that was explained in 4.4.2.2.

From the practical point of view, the "attributes" attached with the VID, *Policies*, $\mathcal{P}$, and $\mathcal{C}$ do not necessarily need to be physically contained within the VID itself. The VID could for instance contain just the pointers to the relevant *Policies* or $Profile$, and be implemented using overlay DHT approach that is commonly used for sharing the information in the P2P or ad-hoc networks in a distributed manner. Nevertheless, the practical implementation of the VID as well as the networking and communication protocols used among the communicating entities are outside the scope of this chapter.

4.4.2.4 Other definitions

Other definitions that are used in the formal specification of CCAAC are as follows:

$$\mathcal{P} = \{Profile_1, Profile_2, \cdots, Profile_n\} \quad (4.11)$$

$$Policies = \{Policy_1, Policy_2, \cdots, Policy_n\} \quad (4.12)$$

$$\mathcal{AR} \in \{Read, Write, NULL\} \quad (4.13)$$

Please note that $\mathcal{P}$ is a set of entity's $Profile$ contained in the VID, and is stored in the Profiles Repository. The same goes for *Policies*, it is a set of any entity's *Policy* and is stored in the Policies Repository. The usage of both Profiles and Policies Repository will be shown when the detail mechanisms of CCAAC will be explained in the next sub section. As for the AR, it can either be $\{Read\}$, $\{Write\}$, $\{Read, Write\}$, or $\{NULL\}$. If $AR = \{NULL\}$, the permission to access a particular object is not allowed.

4.5 Specification of CCAAC Mechanisms

In the following, the formal specification for two of the four main processes in the CCAAC secure access control mechanism is given in details, i.e. the capability creation and access. The specification assumes that CASM is used and the message exchanges among the internal modules are as depicted in Figure 4.1. In the proposed CCAAC model, the SDP is not only acting as a central point for security decisions as was the case in CASM, but is also responsible of performing other functionalities as explained in the following.

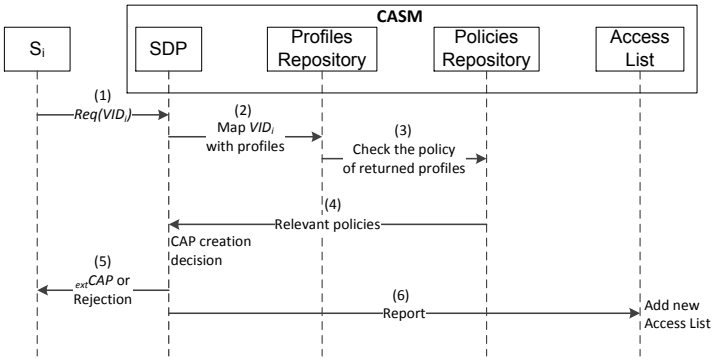

Figure 4.3: Capability creation protocol in the proposed access control mechanism

4.5.1 Creation

The capability creation protocol in the proposed access control mechanism is presented in Figure 4.3 and can be explained as follows:

1. The *Subject* (**S**) sends a capability creation request of a certain *Object* (**O**) along with its own VID (VID_S) and the VID of the **O** (VID_O) to be accessed are sent by **S** to the SDP. As discussed earlier, **S** accesses **O** according the VID_O that can be "seen" by the **O**. Whereas the VID_S will be used later by the **O** to obtain $\mathcal{P}$ and $\mathcal{C}$ related to **S** later on.

2. Upon receiving this request, the SDP asks the Profiles Repository to map the profiles of **S** given its VID_S. This process will return the Profile $\mathcal{P}$ of **S**.

3. The returned Profile $\mathcal{P}$, together with $\mathcal{C}$, are sent to the Policies Repository, to check the relevant *Policies* of the corresponding *Object* **O** (based on its VID_O).

4. The Policies repository gets all relevant *Policies* from the given $\mathcal{P}$ and $\mathcal{C}$ of the **O** of interests represented by its VID_O, and then gives them to the SDP.

5. The SDP combines the received policies with a policy combining algorithm and comes up with a decision whether to create a new capability (CAP) for **S** or not. In case of positive decision, the SDP creates internal capability of the object ($_{in}CAP$) and stores it in the CASM, as well as creating $_{ext}CAP$ for the **S** and then sends it away. In case of negative decision, rejection message will be sent to **S** instead.

6. The SDP sends a report regarding the CAP creation of an *Object* **O** for the *Subject* **S** to the Access Control Servers (ACS) which will be followed by the creation of a new propagation tree.

The specification of the capability creation in CCAAC can be expressed by two pseudo-codes. The first pseudo-code presented in Algorithm 2 requires a specific *ObjType* as the argument and returns either an ${}_{ext}CAP$ to the subject or an error message if certain conditions are not fulfilled. The other pseudo-code, which is presented in Algorithm 3, does not require a specific *ObjType* as its argument and returns either a set of external capabilities or an error message to the subject.

Algorithm 2 Capability creation for a particular object type

procedure CAPCREATION(VID_S, VID_O)
  $\mathcal{P} \leftarrow getProfiles(VID_S)$
  $\mathcal{C} \leftarrow getContexts(VID_S)$
  $Policies \leftarrow checkPolicies(\mathcal{P}, \mathcal{C}, VID_O)$
  $\mathcal{AR} \leftarrow combinePolicies(Policies)$
  if $\mathcal{AR} \neq NULL$ **then**
    if ${}_{in}CAP == 0$ **then**
      ${}_{in}CAP \leftarrow createIntCAP(VID_O)$
    end if
    ${}_{ext}CAP \leftarrow createExtCAP(VID_S, \mathcal{AR}, VID_O)$
    $sendExtCAP$ *to* **S**
  else
    $Send\_Err$ *to* **S**
  end if
end procedure

4.5.2 Access Provision

The capability access protocol in the proposed access control mechanism is presented in Figure 4.4. The access mechanism based on CCAAC is relying fully on the processing of the ${}_{ext}CAP$, i.e. ${}_{ext}CAP$ validation and evaluation of *Contexts* $\mathcal{C}$ within the ${}_{ext}CAP$, thus it results in an efficient access control mechanism. Furthermore, the authentication can also be achieved along with the access control as we will explain later on in the next section. The overall proposed access mechanism in CCAAC can be further explained as follows:

Algorithm 3 Capability creation in the general case

procedure CAPCREATION(VID_S)
  $MapAR < \mathcal{O}, \mathcal{AR} >= [\,]$
  $\mathcal{P} \leftarrow getProfiles(VID_S)$
  $\mathcal{C} \leftarrow getContexts(VID_S)$
  $Policies \leftarrow checkPolicies(\mathcal{P}, \mathcal{C})$
  $MapAR < \mathcal{O}, \mathcal{AR} > \leftarrow combinePolicies(Policies)$
  if $MapAR \neq NULL$ **then**
    for all $\mathcal{O}$ *in* $MapAR$ **do**
      if $\mathcal{AR} \neq NULL$ **then**
        if ${}_{in}CAP == 0$ **then**
          ${}_{in}CAP \leftarrow createIntCAP(\mathcal{O})$
        end if
        ${}_{ext}CAP \leftarrow createExtCAP(VID_S, \mathcal{AR}, \mathcal{O})$
        SendExtCAP to **S**
      else
        SendErr to **S**
      end if
    end for
  end if
end procedure

1. The *Subject* **S** presents its ${}_{ext}CAP_S$ and VID_S to the SDP upon access request.
2. SDP checks the validity of the ${}_{ext}CAP_S$ by running a one-way hash function $f(\mathcal{S}, \mathcal{O}, \mathcal{AR}, Rnd_0)$ and the compare its result with Rnd_i within the ${}_{ext}CAP_S$.
3. In case of a valid ${}_{ext}CAP_S$ being received, SDP will then evaluate a series of $\mathcal{C}_{Const}$ which are contained within the $\mathcal{C}$.
4. The evaluation will return *true* if all the conditions in all of the $\mathcal{C}_{Const}$ are met, otherwise it will return false.
5. The access request response will finally sent back to **S**.

The ${}_{ext}CAP$ validation (step (2) in Figure 4.4) is done by comparing the received Rnd_i within the ${}_{ext}CAP$ with the calculated Rnd_i by means of a one-way hash function $f(\mathcal{S}, \mathcal{O}, \mathcal{AR}, Rnd_0)$. The most important components in the hash function of Rnd_i to be highlighted are the $\mathcal{S}$ and Rnd_0. The $\mathcal{S}$ can be obtained from the $\mathcal{S}$ within the submitted VID_S and the Rnd_0 is obtained from the ${}_{in}CAP$ that is stored within the **O** itself. A secure access control against eavesdropping or replay attacks can be ensured by inserting a nonce within the Rnd_0 as will be shown in Section 4.7.

After the ${}_{ext}CAP$ validation is successfully done, the actual contextual information will be evaluated against *Contexts* $\mathcal{C}$ within ${}_{ext}CAP$ (step (3) in Figure 4.4). It is assumed that the **O**, i.e. device, is able to obtained the actual contextual information from the environment, e.g. sensors or time of access, or by means of messaging mechanisms.

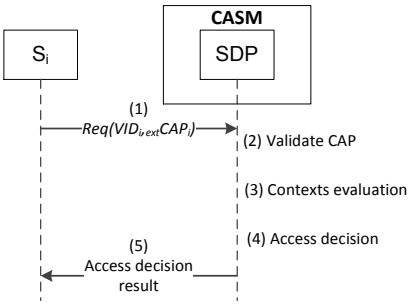

Figure 4.4: Capability access protocol in the proposed access control mechanism

The specification of the access mechanism in CCAAC can be expressed as in the pseudo-code presented in Algorithm 4.

Algorithm 4 Access control with external capability

procedure ACCESS(VID_S, $_{ext}CAP$)
  $AR' = [\,]$
  $Rnd'_i = f(\mathcal{S}, \mathcal{O}, \mathcal{AR}, Rnd_0)$
  if $(Rnd'_i != Rnd_i)$ **then**
    $SendErr$ *to* **S**
  else
    $CtxEval = true$
    for all $\mathcal{C}_{Const}$ *in* $\mathcal{C}$ **do**
      $CtxEval = CtxEval \cap \mathcal{C}_{Const}$
      if $CtxEval$ **then**
        $AR' = AR$
      else
        $AR' = NULL$
      end if
    end for
  end if
end procedure

4.6 Secure CCAAC based Delegation Framework

4.6.1 High level delegation model

To support a federation network in IoT, it is necessary to have an established trust relationships prior to the authority delegation of all the entities involved in this process. The existing solutions in [18] allow all Federation members (in the context of PN) to mutually authenticate to each other, thus establishing trust relationships, by means of key management and security associations mechanisms. By using the similar approach in the context of federated IoT, it is assumed that the trust relationships have been established in all entities joining the federation when it is formed and any new cluster join that federation. Based on this assumption, the proposed high level delegation model can be illustrated as in Figure 4.5. In this figure, *Delegator* is an entity that delegates some or all of its authority to another entity, while *delegatee* is an entity that receives an authority delegation from the *delegator*. According to the first motivating scenario stated in 4.3.2, the user's private device that has the authority to access the information from the smart-fridge is the *delegator*

that delegates its authority partly to the *delegatee* which is the device belongs to retail shop network domain.

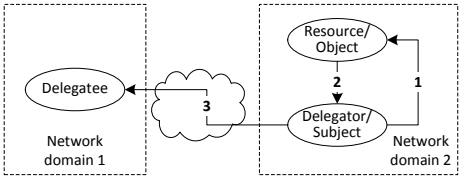

Figure 4.5: High level delegation model on Federated-IoT

Please note that in CCAAC notation, terms such as *Subject* and *Object* is defined. Any *Subject* can be either *Delegator* or *Delegatee*, but *Delegator* will be referred to as *Subject* (**S**) for the sake of the protocol explanation. The resource to be accessed by *Subject* is referred as *Object* (**O**).

A trust relationship is assumed to have been established among all the entities when the federated network is created as discussed extensively in [18]. Therefore, **S** sends a delegation request to **O**. The delegation request is signed with its public key that contains a federated IoT certificate, which is valid in this particular federation, upon requesting its authority delegation towards **D** (step 1 in Figure 4.5). Upon receiving the delegation request from **S**, **O** would verify the signature and then evaluate the delegation request based on some available policies which will be further explained in the next subsection. In case of a positive delegation request evaluation result, a delegation request response in a form of external capability ($_{ext}CAP_D$) with **D**'s identity would be sent to **S**, otherwise an error message would be sent instead (step 2 in Figure 4.5). Finally, the **S** would send the $_{ext}CAP_D$ encrypted with a public key that is known by **D** as a result of trust relationship when federated network between two domains was established (step 3 in Figure 4.5).

4.6.2 Delegation mechanism based on CCAAC

The complete process of the delegation mechanism along with the delegation request evaluation in **O** is depicted in Figure 4.6.

It is important to mention that Figure 4.6 is the micro-level view of Figure 4.5 where **S** and **O** belong to Network Domain 2, and **D** belongs to Network Domain 1. The delegation mechanism depicted in Figure 4.6 is presented as follows:

1. **Sending authority delegation request:** Authority delegation request is being sent by **S** to the SDP within **O**. The request message is signed

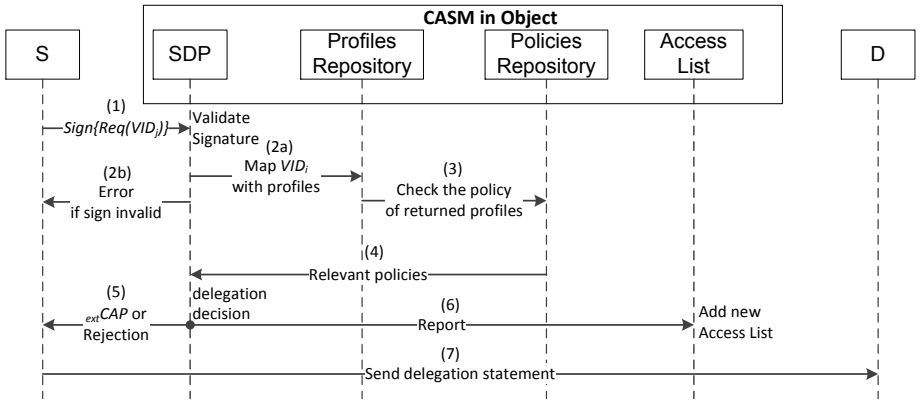

Figure 4.6: Capability propagation protocol for authority delegation in our proposed access control

with **S**'s public that contains a Federated IoT certificate so that **O** is able to make sure the message is indeed sent by **S** and the integrity is maintained. Please note that the type of public key and the specific encryption algorithm being used are not within the scope of this work.

2. **Mapping the VID_D to Profile:** The SDP checks the message's signature. If the signature is valid, the SDP then asks VID-Profiles mapping box to map the profiles of **D** given VID_D. It will return the profile of **D**.

3. **Check the relevant policies:** The returned Profiles $\mathcal{P}$, together with the $\mathcal{C}$ and VID_O, are then sent to the Policies Repository, to check the disclosure policies of the corresponding Object (based on its VID).

4. **Return the relevant policies:** The Policies Repository gets all the relevant policies from the given $\mathcal{P}$ and $\mathcal{C}$ of the object or resource of interest represented by its VID_O, and then gives them to the SDP.

5. **Delegation decision:** The SDP combines the received policies with a policy-combining algorithm and comes up with a decision whether to approve the authority delegation by creating a new capability (CAP) for **D** or not. In the case of a positive decision, the SDP creates a delegation statement in the form of ${}_{ext}CAP$ for the **D** and then sends it to **S**. The difference between the newly created ${}_{ext}CAP$ and the one that is owned by the **S**, lies in the **D**'s identifier $\mathcal{D}$ within the Rnd_i component. In the case of a negative decision, a rejection message will be sent to **S** instead.

6. **Update propagation tree:** In parallel to sending the delegation decision, the SDP sends a report regarding the CAP creation of an object

for Subject i, S_i, to the ACS which will be followed by the creation of a new propagation tree.

7. **Sending authority delegation statement:** Finally, **S** sends the authority delegation statement in the form of ${}_{ext}CAP$ particularly for **D**. Moreover, in order to maintain the confidentiality and integrity of the ${}_{ext}CAP$, it can be signed with a shared secret key between **S** and **D**, based on an assumption that both domains have established a trust relationship by authenticating each other through a certain key pair.

Steps 2 through 5 is expressed in the pseudo-code in Algorithm 5.

Algorithm 5 Capability delegation decision

procedure DELEGATECAP(VID_D, VID_O)
  $\mathcal{P} \leftarrow getProfiles(VID_D)$
  $\mathcal{C} \leftarrow getContexts(VID_D)$
  $Policies \leftarrow checkPolicies(\mathcal{P}, \mathcal{C}, VID_O)$
  $decision \leftarrow combinePolicies(Policies)$
  if $decision \neq NULL$ **then**
    if $IntCAP = 0$ **then**
      ${}_{in}\mathcal{CAP} \leftarrow createIntCAP(VID_O)$
    end if
    ${}_{ext}\mathcal{CAP} \leftarrow createExtCAP(VID_D, \mathcal{AR}, VID_O)$
  end if
end procedure

It is assumed that **S** as the *delegator* knows the identity of the **O** and **D**. This is possible when the *delegator* **S**, subscribes to a service, in which a device in the service provider's domain needs to be given an authority delegation, i.e. as *delegatee* **D**, in order to access a device or resource within the *delegator*'s network domain, i.e. **O**. With this assumption, **S** needs to submit a delegation request by stating the identities of **D** and **O** in the form of VID_D and VID_O, respectively. Once submitted, *delegatee*'s Profile ($\mathcal{P}$) and Context ($\mathcal{C}$) can be obtained from VID_D as they are attached to it (see Equation 4.10). Afterwards, all relevant policies related to VID_O that contain *delegatee*'s $\mathcal{P}$ and $\mathcal{C}$ are gathered from the Policies Repository to be further evaluated by a certain Policies Combining Algorithm to obtain a delegation decision.

4.7 Evaluation and analysis

The evaluation will focus on secure capability creation and access mechanisms as the most important processes in the access control, especially when capability is involved. In order to secure the access control mechanism, simple mechanisms of generating nonce in both sides and a secret key cryptography to encrypt the message are introduced. The Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [12] which is based on the Dolev-Yao [4] model is used for model verification purposes as well as for evaluating the secrecy and authentication between the subject, i.e. the one that requests access, and the object, i.e. the one that is being accessed.

4.7.1 Evaluation procedure

In order to carry out the evaluation using AVISPA, the following assumptions are made:

- Both *Subject* and *Object* have already obtained shared key(s), i.e. either symmetric or asymmetric, through key generation and sharing mechanisms prior to the capability creation process.
- To prevent a replay attack, a nonce is incorporated within the Rnd_0 of ${}_{in}CAP$ and ${}_{ext}CAP$.
- A new key, which is a result of a one-way hash function using nonce generated by both parties, is presented by **S** that wants to access **O**.
- The access mechanism using external capability is used immediately after the capability creation in order to have an integrated evaluation of both mechanisms.

The complete protocol evaluation is presented in the following model:

$\mathbf{S} \rightarrow \mathbf{O}$: $\{\mathcal{S}.Req.N_s\}\_K_{so}$
$\mathbf{S} \leftarrow \mathbf{O}$: $\{\mathcal{S}.\mathcal{O}_{xo}.\mathcal{AR}_s.\mathcal{C}_s.N_o.Rnd_i\}\_K_{so}$
$\mathbf{S} \rightarrow \mathbf{O}$: $\{\mathcal{S}.\mathcal{O}_{xo}.\mathcal{AR}_s.\mathcal{C}_s.Rnd_i\}\_K_N$

where

- $\mathcal{S}$: Subject identifier.
- $\mathcal{O}$: Object identifier.
- $\{\ \}\_$: A symbol of encryption.

- *Req*: A capability creation request message.
- N_s: A nonce generated by **S**.
- N_o: A nonce generated by **O**.
- K_{so}: A shared secret key shared by the **S** and **O** prior to the capability creation process to encrypt the whole message.
- K_N: A new shared secret key generated by the **S** as a result of a one-way hash function, $f(N_s.N_o)$.
- Rnd_i: A result of a one-way hash function $f(\mathcal{S}.\mathcal{O}_{xb}.\mathcal{AR}_s.\mathcal{C}_s.Rnd_0)$.
- Rnd_0: A result of a one-way hash function $f(\mathcal{O}_{xb}.\mathcal{AR}_s.N_o)$.

Besides the protocol that involves **S** and **O**, an intruder, **I**, based on Dolev-Yao intruder model has been introduced in the evaluation. The intruder **I** is assumed to have the knowledge of the following:

- $\mathcal{S}$: the Subject's identifier.
- K_{si}: the shared secret key between **S** and **I**.
- K_{io}: the shared secret key between **I** and **O**.
- all the hash functions being used in the protocol, i.e. $f(N_s.N_o)$, $f(\mathcal{S}.\mathcal{O}_{xo}.\mathcal{AR}_s.\mathcal{C}_s.Rnd_0)$, and $f(\mathcal{O}_{xo}.\mathcal{AR}_s.N_o)$.

The goal of the evaluation is to verify the secrecy of the new generated key presented by **S**, e.g. K_N, in order to access **O**, and the N_o generated by **O** in order to create Rnd_0. Furthermore, N_o is also used to authenticate **S** over **O**. These two important aspects in the evaluation will be discussed further in the coming subsections.

4.7.2 The secrecy

The secrecy of N_o is an important component to prevent replay attacks on the ${}_{ext}CAP$ that is submitted by **S**. The evaluation result using the AVISPA tool shows that the secrecy of N_o is kept. It is important to note that the Dolev-Yao intruder model used in the evaluation is assumed to have knowledge of Subject's identifier $\mathcal{S}$, the hash functions being used, and the shared keys, K_{si} and K_{io}. The intruder cannot decrypt the message being sent by either side even if he can capture it since he has no knowledge of K_{so}. In other words, the knowledge of K_{si} and K_{io} that might be used by the intruder to fool both sides

will be useless in this case. As a result, both nonce, i.e. N_s and N_o, remain secret. Secondly, the secrecy of K_N which is a result of simple one-way hash function $f(N_s.N_o)$ as presented in the evaluation model is another important fact to be evaluated. Similar with the N_o, the evaluation result by AVISPA tool also shows that K_N remains secret. As explained previously, the secrecy of K_N is a result of the secrecy of both N_s and N_o due to inability of the intruder to decrypt the message without having the knowledge of K_{so}. Based on this evaluation result, i.e. the fact that K_N remains secret, it could be concluded that the ${}_{ext}CAP$ is safe from eavesdropping and replay attacks.

4.7.3 The authentication

The authentication of **S** against **O** can be achieved when **O** receives a valid Rnd_i. In particular, the received N_o within the Rnd_i, which is supposed to be a secret nonce generated by **O** and known only by **S**, has to be valid and kept secret. The evaluation result by AVISPA tool shows that authentication is achieved as well. As stated previously in the secrecy analysis, the fact that the secrecy of N_o that can be ensured in the protocol results in the validity of Rnd_i within the ${}_{ext}CAP$, thus **S** can successfully authenticated by **O**. Furthermore, this result also shows that the proposed CCAAC model is not only able to provide access control but also authentication at the same time. Hence, the main objective of providing security mechanisms, i.e. authentication and access control, in order to prevent security threats, especially in the context of IoT, has been achieved.

4.8 Conclusion

Access control is of paramount importance for a full thrive of IoT, especially due to the dynamic network topology and distributed nature. In this chapter, different access control models with their advantages and limitations have been discussed. Based on this, the secure CCAAC model with specifications of the capability (CAP) creation and access mechanisms has been proposed. The definition of Federated IoT is used as a baseline in designing the proposed authority delegation mechanism that is part of the overall CCAAC model. The protocol description and security consideration involving the usage of cryptographic keys are further presented in the chapter to give some guidelines in the practical implementation. The proposed CCAAC has been analysed in the presence of security threats, such as eavesdropping and replay attack by an intruder, in order to test its resilience. Security proofs and evaluations by using

AVISPA tool show that the CCAAC model achieves not only access control but also the secrecy of CAP and authentication.

The are several works remain to be done in this research area. First, incorporating the CCAAC model with light-weight authentication supporting capability, such as the one proposed in [17], and to evaluate it with a more realistic adversaries model. Second, the current authority delegation framework in the proposed CCAAC assumes trust relationships that are already established among entities in different federation domain. The future outlook will consider the case in which no prior knowledge of the trust relationship between two network domains in Federated IoT. Unlike the approach used in this chapter, an additional entity that is trusted by both domains, for instance Identity Provider (IdP), needs to be involved in the design. Lastly, prototype implementation of all aspects in CCAAC by using the existing access control realization model or on a device level will be something to look forward.

4.9 References

[1] Bayu Anggorojati, Parikshit N. Mahalle, Neeli R. Prasad, and Ramjee Prasad. Capability-based access control delegation model on the federated iot network. In *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, September 2012.

[2] Tuomo Repo Arto Hämäläinen, Jari Porras and Pekka Jäppinen. Applying wireless technology to access control systems. In *1st Workshop on Applications of Wireless Communications (WAWC'03)*, 2003.

[3] Tyler Close. Acls don't. Technical report, Hewlett Packard Laboratories, 2009.

[4] D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198 – 208, mar 1983.

[5] Patrik Floréen, Michael Przybilski, Petteri Nurmi, Johan Koolwaaij, Anthony Tarlano, Matthias Wagner, Marko Luther, Fabien Bataille, Matthieu Boussard, Bernd Mrohs, and Sian Lun Lau. Towards a context management framework for mobilife. In *IST Mobile & Communications Summit*, 2005.

[6] H. Gomi. Dynamic identity delegation using access tokens in federated environments. In *Web Services (ICWS), 2011 IEEE International Conference on*, pages 612 –619, july 2011.

[7] Hidehito Gomi, Makoto Hatakeyama, Shigeru Hosono, and Satoru Fujita. A delegation framework for federated identity management. In *Proceedings of the 2005 workshop on Digital identity management*, DIM '05, pages 94–103. ACM, 2005.

[8] L. Gong. A secure identity-based capability system. In *Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on*, pages 56 –63, may 1989.

[9] M. Grossmann, M. Bauer, N. Honle, U.-P. Kappeler, D. Nicklas, and T. Schwarz. Efficiently managing context information for large-scale scenarios. In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 331 –340, march 2005.

[10] K. Hasebe and M. Mabuchi. Capability-role-based delegation in workflow systems. In *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on*, pages 711 –717, dec. 2010.

[11] Koji Hasebe, Mitsuhiro Mabuchi, and Akira Matsushita. Capability-based delegation model in rbac. In *Proceeding of the 15th ACM symposium on Access control models and technologies*, SACMAT '10, pages 109–118, New York, NY, USA, 2010. ACM.

[12] http://avispa project.org/.

[13] Young-Gab Kim, Chang-Joo Mon, Dongwon Jeong, Jeong-Oog Lee, Chee-Yang Song, and Doo-Kwon Baik. Context-aware access control mechanism for ubiquitous applications. In Piotr Szczepaniak, Janusz Kacprzyk, and Adam Niewiadomski, editors, *Advances in Web Intelligence*, volume 3528 of *Lecture Notes in Computer Science*, pages 932–935. Springer Berlin / Heidelberg, 2005.

[14] Devdatta Kulkarni and Anand Tripathi. Context-aware role-based access control in pervasive computing systems. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, SACMAT '08, pages 113–122, New York, NY, USA, 2008. ACM.

[15] Henry M. Levy. *Capability-Based Computer Systems.* Butterworth-Heinemann, 1984.

[16] Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson. Rapid prototyping of mobile context-aware applications: the cyberguide case study. In *Proceedings of the 2nd annual international conference on Mobile computing and networking*, MobiCom '96, pages 97–107, New York, NY, USA, 1996. ACM.

[17] Parikshit N. Mahalle, Bayu Anggorojati, Neeli R. Prasad, and Ramjee Prasad. Identity establishment and capability based access control (iecac) scheme for internet of things. In *Wireless Personal Multimedia Communications (WPMC), 15th International Symposium on*, September 2012.

[18] R. Prasad. *My personal Adaptive Global NET (MAGNET).* Signals and Communication Technology Book. Springer Netherlands, 2010.

5

Capability-based access control in the M2M local cloud platform: A practical implementation perspective

Protection and access control to resources plays a critical role in a distributed computing system like M2M and cloud platform. The M2M local cloud platform considered in this paper, consists of multiple distributed M2M gateways that form a local cloud – presenting a unique challenge to the existing access control systems. The most prominent access control systems, such as ACL and RBAC, lack in scalability and flexibility to manage access from users or entity that belong to different authorization domains, and thus unsuitable for the presented platform. The access control approach based on API keys and OAuth that is used by the existing M2M Cloud platform, fails to provide fine grained and flexible access right delegation at the same time when both methods are used together. The proposed approach is built upon capability-based access control that has been specifically designed to provide flexible, yet restricted, access rights delegation. A number of use cases are provided to show the usage of capability creation, delegation, and access provision, particularly in the way application accesses services provided by the platform.

5.1 Introduction

The concern about security and privacy play a huge role for mass adoption of the IoT/M2M as cloud services [8]. More specifically, access control to information and resources is very critical to ensure security and privacy of a system. The M2M local cloud platform is a platform that consists of distributed M2M gateways that forms a cloud locally and provides M2M services collectively. Consequently, some unique challenges in access control is imposed, especially in the scalability and flexibility aspects.

ACL is the most common way of providing access control, but it becomes more and more complex as the number of subjects and resources increases. The RBAC [9] is introduced to cope with this issue, by mapping subjects into roles. Many access control models are also derived based on RBAC model. However, it cannot cope with the problems of role explosion and multi administrative domains. The ABAC system tries to solve the roles explosion issue by specifying access policies and rules based on the attributes of the subjects and resources, thus potentially reducing the number of rules. However, defining a consistent attributes is not an easy task, especially in the M2M world where the context is highly dynamic. Overall, all previously mentioned access control mechanisms have a common problem which is difficulties in enforcing least privilege access principle and scalability problem which is a major requirements in M2M. Additionally, they do not provide flexible access rights delegation feature.

One of the existing IoT/M2M Cloud platform, *Xively*, relies on the typical access control solution in many cloud services. The API keys can be created for the devices, client applications, and actual users to authorize the access of specific resource in *Xively* [10]. However, this API key is managed centrally in *Xively* cloud platform and is not delegable. On the other hand, OAuth [7] is implemented in *Xively* to allow access of third party application to resources owned by a user. However, using the API key in the OAuth application would mean granting access to all resources tied with the application to the third party.

A capability-based access control have been seen as a promising solution to problems in IoT/M2M by some recent research works, but its actual implementation is still a big challenge. The CCAAC presented in Chapter 4, tries to solve the above mentioned issues in the context of IoT using the capability-based access control, but the use of access policy based on VID is not clear and quite impractical to be implemented. Additionally, the attempt to highly control the capability propagation significantly reduce the flexibility of access

right delegation. Another capability-based access control in IoT has been done in [4] in which the rights delegation and revocation based on capability have been emphasized. However, it does not explain how the capability is requested and granted, which is the most important part especially in the IoT and M2M world. The proposed solution is built upon capability-based access control designed to provide flexible, yet restricted, access rights delegation.

The rest of this chapter is organized as follows: An overview of the M2M local cloud platform architecture and the way applications access the platform is described in Section 5.2. The detail of the proposed approach is presented in Section 5.3. Some use cases explaining how the proposed approach is applied in the real platform is explained in Section 5.4. Finally, the conclusion and future outlooks are given in Section 5.5.

5.2 System Description

Different approach in the IoT or M2M architectures, as reported in [8], will impact the security challenges and their solutions. As the access control mechanism presented in this paper is particularly designed and developed for the M2M local cloud platform within the European FP7 BETaaS (Building the Environment for the Things as a Service) project, thus it is necessary to fulfil the unique technical requirements and architecture of this platform. This section is dedicated to explain the BETaaS platform's architecture, and the mechanism in which the applications and users can access the services provided by BETaaS platform. It is important to note that although the access control mechanism is designed and developed for BETaaS project, it is also applicable to other platforms, especially the M2M and Cloud platforms.

5.2.1 BETaaS architecture

BETaaS is *a content-centric platform distributed over a local cloud, hosted by gateways, providing an environment for applications accessing M2M services and devices through a set of services* [6]. In principle, the platform consists of multiple distributed M2M gateways (GW)s that are collaborating to each other and forming a so called *local cloud* or *BETaaS instance*. With this approach, there can exist different BETaaS instances which do not necessarily connected to each other. This approach is totally different than the typical M2M deployment where the cloud is hosted in a centralized cloud provider.

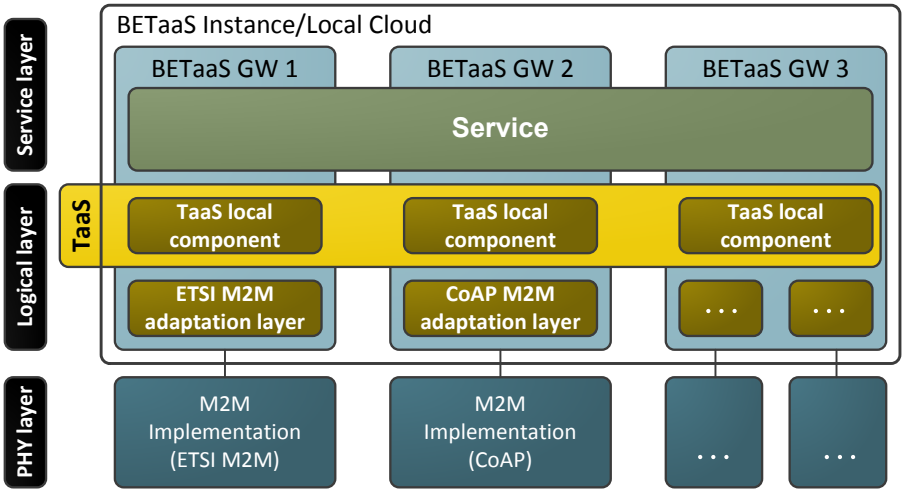

Figure 5.1: High level architecture of the BETaaS platform

The high level architecture of BETaaS platform consists of three layers: *Service* layer, *Logical* layer, and *Physical* layer, as illustrated in Fig. 6.1. On the practical deployment perspective, the main functionality of the platform resides in the *logical* and *service* layers, which are implemented in a GW. It is important to note that since a GW is implemented in a device more likely to have very minimum user intervention, the access control mechanism should work in the same manner also.

At the functional architecture point of view, BETaaS consists of several *managers* that perform different functionalities, one of it is the security manager [2]. The layered structure as in the high level architecture is also applied in the design of the *managers*, e.g. security manager. The purpose of such design is to allow clear division of security tasks in different layer, e.g. application level security resides in the service layer, the GW level security is at the Thing as a Service (TaaS) sub-layer, and the adapation sub-layer handles things level security. Please note that in the context of BETaaS, thing(s) refers to any device or object with communication capability, such as sensor, actuator, tag, etc.

5.2.2 Security manager

The security manager consists of some sub-managers, namely Key, Authentication, Authorization, and Trust managers, which perform functionalities according to their respective names. Brief description of each sub-manager

of the proposed security manager in BETaaS which are closely related to the proposed access control mechanism is presented in the following sub-sections.

5.2.2.1 Key Manager

Key management is a very important functionality which defines how to manager the secret (and shared) keys, as the important components to perform any security operation. The key management has a role to manage the associations of different entities (e.g. GWs with an instance, application with an instance, etc), which will be used to perform authentication, and later on to manage application and user sessions as well as to perform encrypted communication.

A common security mechanism to provide trust relies on the PKI in a form of a digital certificate issued by the trusted third party, i.e. CA. In some cases however, the BETaaS instance does not have a constant access to the external CA, e.g. local cloud that is built upon ad-hoc network, the method which involves external CA cannot be directly applied. Therefore, a key management mechanism that relies on local certificate issued by an *internal* CA is proposed.

In the context of BETaaS, the *internal* CA refers to a gateway which acts as a common CA of that particular instance, also known as GW*. At the beginning of BETaaS instance creation, the GW* would initiate the creation of root and intermediate certificates which forms the root credential of this particular instance. For a new GW to join the instance, it needs to generate a key pair, i.e. public and private keys, and then sends a join request to GW* along with the generated public key. Upon receiving join request, GW* would create a credential for the new GW which includes a certificate generated from the public key and other submitted information by the new GW and is signed using the private key of the GW*'s credential. This new created credential plays an important role to provide trust relationship among all the GWs within the local cloud or instance, including the access control mechanism as well as in deriving key for establishing secure communication between two GWs.

The ECC based PKI is used in the practical implementation of key manager because it requires smaller key size compared to other crypto algorithm, such as RSA, to provide the same level of security. For example, a 160 bit ECC is equivalent to a 1024 RSA, a 192 bit ECC is equivalent to a 1536 bit RSA, a 224 bit ECC is equivalent to a 2048 bit RSA and so on. A set of European Commission (EC) domain parameters recommended by National Institute of Standards and Technology (NIST) and Standard of Efficient Cryptography (SEC) are chosen in order to generate a key pair of ECC.

5.2.2.2 Authentication Manager

In the context of BETaaS, two level of authentications are considered, namely gateway level authentication and application and service level authentication.

The gateway level authentication mainly deals with the mutual authentication in order to establish secure communication between two GWs. The method to perform mutual authentication is based on the Elliptic Curve Menezes-Qu-Vanstone (ECMQV) protocol. ECMQV is a three-pass key agreement protocol that has been standardized in ANSI X9.63, IEEE 1363, ZigBee Smart Energy 1.0, etc [5]. In principle, it is an extension of the ordinary Diffie-Hellman key agreement protocol with the ECC. Among the benefits of ECMQV are lower computational complexity and bandwidth reduction. Detail explanation of how ECMQV protocol works can be reviewed in [5].

The application and user level authentication also relies on the ECC based PKI. But unlike the key management mechanism that relies on local certificate issued by the internal CA, an external CA is used to issue certificate for application. The certificate for application is obtained through a registration process and the authentication by means of certificate verification only occurs upon application installation in the platform. Further explanation about the application authentication and installation process is presented in Section 5.2.3.

5.2.2.3 Authorization Manager

Authorization manager is the central component that performs access control mechanism in the platform. The detail explanation of how the access control mechanism within the authorization manager works and how it interacts with other components will be presented in Section 5.3 and 5.4.

5.2.2.4 Trust Manager

In the context of BETaaS, trust represents the level of reliability on a Thing or Thing Service to produce certain data and/or perform certain actions, in the expected conditions. In the end, this is related to the reliability trust, since it is assumed that the system relies on a set of Things in order to execute services in BETaaS platform. However, the detail explanation of trust manager is out of the scope of this chapter because it is totally independent with the proposed access control mechanism.

5.2.2.5 Interaction with other managers in BETaaS platform

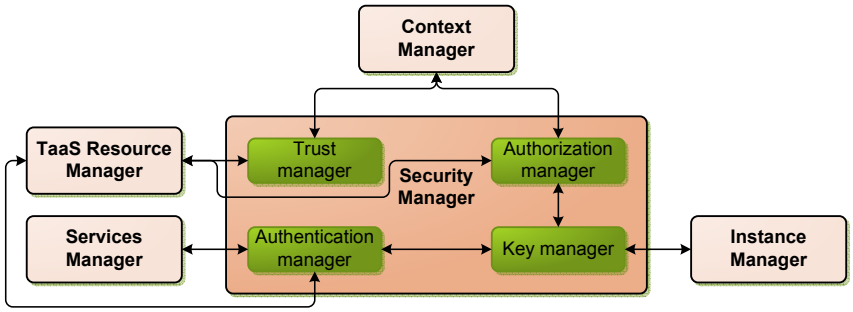

Figure 5.2: The interaction of the Security Manager with the other Managers

Figure 5.2 shows the interaction among the sub-managers in the security manager with the other managers in BETaaS platform. The Instance Manager is responsible for GW joining and un-joining process, thus some interaction with the Key Manager is required in order to acquire a certain long term key to prove that the GW is part of the local cloud. The context manager is responsible to provide contextual information and semantic data, e.g. location information, sensor and actuator data, etc, to the platform; including the trust manager which requires this information to calculate the trust score of all thing(s) that are connected with the local cloud, and authorization manager that may need to make an access decision based on a certain context information. The service manager is a component in BETaaS that has a direct contact with the external application. In the practical API design, the only interaction between service manager and security manager is at the beginning of application installation process, when the application presents its API key to be validated by the authentication manager. Lastly, the TaaS resource manager is a central component in the operation of BETaaS GW, thus it has all the interfaces to the security manager sub-components. The TaaS resource manager also acts as a bridge between security manager and service manager in most of the API interactions.

5.2.3 Access to BETaaS platform

The M2M services exposed by BETaaS platform can be accessed by users through applications. The term application in BETaaS – also in other cloud platform – is considered as a process outside of the platform that can consume the services provided by any BETaaS platform through BETaaS APIs. In the typical cloud service deployment, including the M2M Cloud platform, an

application obtains a set of API keys that allows it to access different set of services provided by the Cloud platform. These keys are normally generated during the registration process, e.g. the application developer would register its application in the cloud platform before starting the development process. However, the same approach cannot be applied in BETaaS due to multiple BETaaS instances or local clouds with different owners that may exist in all over the world, while implementing the same sets of APIs.

On the other hand, such a distributed cloud deployment also impacts the way an application accesses different BETaaS platform. For example, the same intrusion detection application developed for BETaaS platform will not necessary be able to consume the same service through the same API from a BETaaS instance deployment in the same way as those in the other BETaaS instances deployment. That is due to the availability of a particular thing service and other requirements that pertaining to each independent local cloud. To cope with this issue, an installation procedure is needed when an application accesses a BETaaS instance for the first time [2]. The purpose of this procedure is for allocating the resources required for the application according to a set of requirements in the local cloud. The authenticity check of the application is also performed during the application installation. It is based on the typical PKI where the CA can be any external trusted third party (unlike the key management at the GW level), such as BETaaS application store. However, relying on the authenticity check in the installation procedure and the typical API keys solution are still unable to solve the access control issue, let alone the issues of scalability and flexibility in access rights delegation that are inherent in the existing access control mechanisms, which have been pointed out in Section 5.1.

5.3 The proposed approach

The proposed approach is based on the capability based access control. Some enchancements are made in order to the problems faced by the other capability based access controls. This section explains the overall capability design along with the mechanisms involved in the capability based access control.

5.3.1 Capability design

The proposed capability consists of internal and external capability, e.g. ${}_{in}CAP$ and ${}_{ext}CAP$ respectively, similar to the basic use of capability in our previous work [1]. In BETaaS's context, the resource or object is the *thing service* which is owned by a GW, thus ${}_{in}CAP$ is created and owned by any GW that offers *thing services*. *Thing service* refers to a service provided by a *thing*, e.g. temperature sensing service by a temperature sensor. The ${}_{ext}CAP$ can be obtained by any party that wish to access a resource that corresponds to the ${}_{in}CAP$, including another GW in the local cloud, applications, and users.

The basic structure of ${}_{in}CAP$ is as follows:

$$ {}_{in}CAP = \{\mathcal{O}, DSign\} $$

where

- $\mathcal{O}$: The name of *object* or *resource* to be accessed.
- $DSign$: A *digital signature* of this ${}_{in}CAP$, which is obtained by signing the hash of the ${}_{in}CAP$'s content, i.e. $\mathcal{O}$, with GW's private key that is obtained by the key management mechanism in Section 5.2.1. It is used to check the integrity of the capability's content.

On the other hand, ${}_{ext}CAP$ contains much more information than the ${}_{in}CAP$ which is necessary for the capability delegation and revocation mechanisms.

$$ {}_{ext}CAP = \{\mathcal{I}, \mathcal{S}, \mathcal{AR}, \mathcal{O}, \mathcal{VC}, \mathcal{R}, DSign\} $$

where

- $\mathcal{I}$: The information about the *issuer* of this ${}_{ext}CAP$.
- $\mathcal{S}$: The information about the *subject* or the ${}_{ext}CAP$ holder.
- $\mathcal{AR}$: The details of *access rights* over the object or resource granted to the subject.
- $\mathcal{O}$: The name of *object* or *resource* to be accessed.
- $\mathcal{VC}$: The *validity condition* of the ${}_{ext}CAP$, e.g. validity period.
- $\mathcal{R}$: The URL of the capability *revocation* service.

- *DSign*: A *digital signature* of this ${}_{ext}CAP$, which is obtained by signing the hash of the ${}_{ext}CAP$'s content with issuer's private key.

$\mathcal{I}$ is a data structure that contains {*IssuerType, IssuerCertificate, IssuerCapabilities*}. The $IssuerType$ can either be a *GW, application*, or *user*. The $IssuerCertificate$ is the certificate obtained by a GW upon joining a local cloud, or by an application upon the registration (i.e. before using the APIs of the cloud services in the application), or by a user upon the registration (i.e. within the local cloud platform or the individual GW). The $IssuerCertificate$ field plays an important role in the access delegation because the validity of a delegated capability depends on the validity of the issuer's certificate. In addition, it is also used to check the validity of the digital signature ($DSign$) field of this capability. $IssuerCapabilities$ is the chain of capabilities which shows the access delegation chain that the issuer hold.

$\mathcal{S}$ is another data structure that contains {*SubjectType, SubjectPublicKeyInfo*}. Similar to $IssuerType$, the $Subject\ Type$ can be a *GW, application*, or *user*. The $SubjectPublic\ KeyInfo$ is an optional field that gives an information about the public key of the subject, i.e. capability's holder.

$\mathcal{AR}$ is also a data structure that contains the following field: {*AccessType, AccessConditions*}. The *AccessType* is either *GET, PUT, POST* or *DELETE*, which is equivalent to *Read, Write, Modify* or *Delete*. The *AccessConditions* states some access conditions to be excercised upon granting the access permission, which may include some context information.

5.3.2 Capability creation

Initially, the object or resource owner, i.e. GW, creates ${}_{in}CAP$s for all the objects that it have authority over. In BETaaS context, a ${}_{in}CAP$ is created when a *thing* that provides *thing services* is connected to the GW. The ${}_{ext}CAP$ can then be granted upon request by the external parties that wish to gain access to the a resource that coresponds to that particular ${}_{ext}CAP$.

In principle, the detail mechanism of the ${}_{ext}CAP$ creation is similar to the capability creation presented in Chapter 4 with the exeption that no VID is used in this approach. Rather, it uses a digital certificate that contains a GW's or application's public key, that is signed by the GW* or by the BETaaS Apps's trusted CA respectively. In addition, a set of access policies is created and updated by the GW's owner, thus allowing a fine tuned access that fits the needs and requirements of each individual GW in a local cloud.

5.3.3 Access right delegation mechanism

The access right can be easily delegated by propagating the ${}_{ext}CAP$, but it is difficult to control its propagation [3]. Some attempts have been made to control the capability delegation in the previous works. Our previous work uses a mechanism where the delegator needs to ask the original resource owner before delegating its ${}_{ext}CAP$ which reduces the flexibility of the access rights delegation through capability [1]. On the other hand, [4] tries to control the right delegation by specifying the delegation depth in the access right field within the capability. However, it is impractical to predict the proper depth of delegation in advance as one can always need to go one more level down.

In the proposed approach, it is assumed that the ${}_{ext}CAP$ is delegable by default, i.e. no need to specify the *delegable* and *delegation depth* as in [4]. Anybody can delegate a subset of its access rights stated in the ${}_{ext}CAP$ by creating another ${}_{ext}CAP$, specifying its ID, certificate, and a list of ${}_{ext}CAP$ chain in the issuer field, and sign the ${}_{ext}CAP$ with its own private key. When the delegatee, i.e. the ${}_{ext}CAP$ holder that receives a delegated access, accesses the resource and presents its ${}_{ext}CAP$ to resource owner for the first time, the resource owner will evaluate the ${}_{ext}CAP$. The most critical point in the evaluation that controls the access right delegation is by validating the ${}_{ext}CAP$ issuer based on the issuer's certificate (for GWs and applications) or user credential (for the registered user in the corresponding GW). If the certificate or credential cannot be validated, it simply means that the issuer has no right to delegate the ${}_{ext}CAP$, thus access request is rejected immediately. Otherwise, the ${}_{ext}CAP$'s signature is validated based on issuer's public key to ensure that the delegated ${}_{ext}CAP$ has not been tampered either by the delegatee or other parties. The evaluation also includes all the validity check that is performed in the capability access evaluation (see Section 5.3.4), both for the delegatee's and the delegator's ${}_{ext}CAP$s. The evaluation process also makes sure that the $\mathcal{AR}$ and $\mathcal{VC}$ specified in the delegatee's ${}_{ext}CAP$ are the subsets of those in the delegator's ${}_{ext}CAP$. If the ${}_{ext}CAP$ passes all the previously mentioned validity checks, the resource owner can optionally check the access policies and renew the delegated access rights. Finally, a new ${}_{ext}CAP$ signed by the resource owner's private key will be granted to the delegatee, thus allowing a simpler validity check as a normal ${}_{ext}CAP$ in the next access time.

In this way, the access right delegation mechanism in the proposed approach allows a more flexible access right delegation as compared to the approach in Chapter 4, while providing more control over capability delegation as compared to [4], and maintaining a finer-grained access control as compared to the approach used in [10].

5.3.4 Capability access evaluation

Upon access request by receiving the ${}_{ext}CAP$, the resource owner validates the submitted ${}_{ext}CAP$ by checking its validity period, validating the ${}_{ext}CAP$'s $DSign$, and definitely checking whether the requested access right is listed in the ${}_{ext}CAP$. In addition, the resource owner also checks the validity of the ${}_{in}CAP$ and/or ${}_{ext}CAP$ which are specified in the *IssuerCapabilities* field, and whether or not the ${}_{ext}CAP$ has been revoked through the revocation service. To make it even more secure, the ${}_{ext}CAP$ holder can also sign the ${}_{ext}CAP$ and submit it along with its public key information to prove that he/she really is the ${}_{ext}CAP$ holder, e.g. in case the ${}_{ext}CAP$ is stolen.

5.3.5 Capability revocation

The delegator or ${}_{ext}CAP$ issuer can also revoke a set of ${}_{ext}CAP$s delegated earlier for many different reasons, for instance due to maliciousness, stolen ${}_{ext}CAP$, etc. In principal, this mechanism is the same as certificate revocation, but it is much simpler since the capability is only used in a single place, i.e. the resource owner. The delegated capability can be revoked by sending a capability revocation request to the URL specified in the $\mathcal{R}$ field by mentioning the ${}_{ext}CAP$'s ID and reason of revocation. It is important to mention the reason of revocation because it can determine whether the further delegated ${}_{ext}CAP$ should be revoked or not.

5.4 Use cases

Some use cases will be presented to show how the proposed capability-based access control work in different operations of the M2M local cloud, e.g. GW join to local cloud, application installation, and other delegation cases. For the sake of simplicity, the presented use cases only show the security aspects of the actual operations, i.e. other aspects like resource allocation, context and QoS evaluations are not completely shown.

5.4.1 GW joining process

In Fig. 5.3, the GW joining process is shown. In this process, the new GW that wants to join to the local cloud sends a join request to the GW* by including its long term public key and some information about it, e.g. user

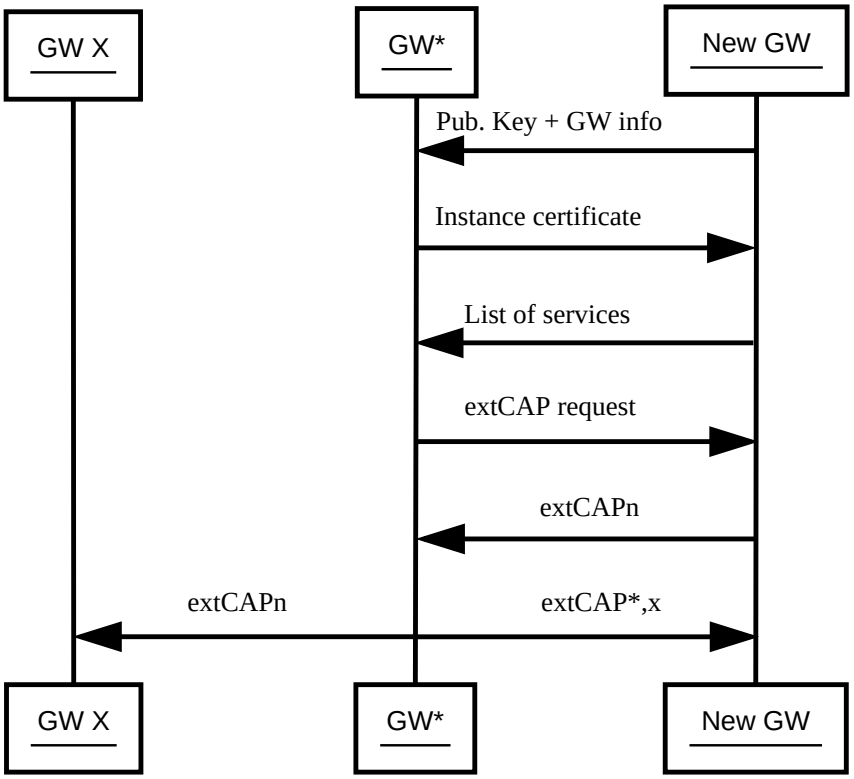

Figure 5.3: Capability creation and delegation during GW joining process

friendly name, address, etc. If the join request is accepted, GW* will issue an instance certificate for the new GW that is used to provide trust among all GWs member of the local cloud. Upon joining the local cloud, the new GW is also required to present the list of services it offers to all other members of the local cloud through the GW*, such that all GWs can synchronize the existing services with the new services provided by new GW. Upon receiving list of services from the new GW, GW* sends an ${}_{ext}CAP$ creation request over all the services or resources provided by the new GW. Upon receiving all the ${}_{ext}CAP$s from the new GW, i.e. ${}_{ext}CAP_n$, GW* will then delegate the ${}_{ext}CAP_n$ to the remaining GWs (if there is one) and also delegate the ${}_{ext}CAP$s of all the other GWs including itself, i.e. ${}_{ext}CAP_X$ and ${}_{ext}CAP_*$ to the new GW. Please note that each ${}_{ext}CAP$ is signed by the private key of each GW that provide its services, and this signature can be validated by another GW through the public key (as pair of the signing key) that can be extracted from the instance certificate attached in the $\mathcal{I}$ of the ${}_{ext}CAP$. In this way, the receipient of the ${}_{ext}CAP$, both direct and delegated recipients, can verify it safely. In addition, the service provider GW can also verify the issuer of the ${}_{ext}CAP$, upon receiving an access request by another GW that obtains the

$_{ext}CAP$ by means of delegation. It also means that another GW is able to further delegate access right to external application or users.

5.4.2 Application installation

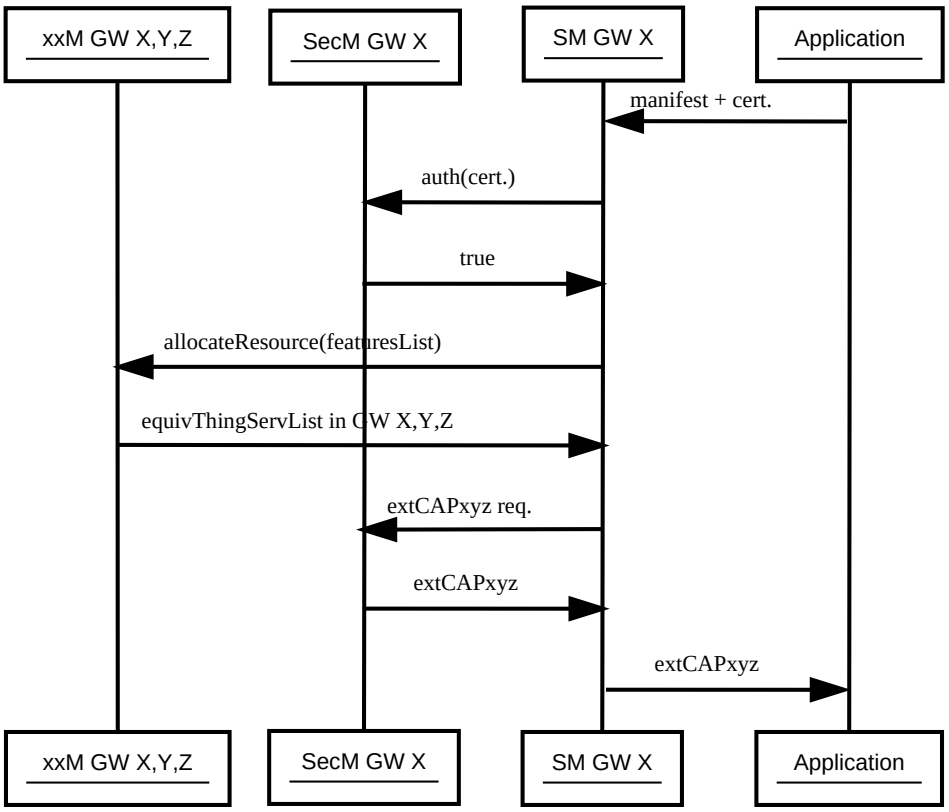

Figure 5.4: Capability delegation during application installation process

Fig. 5.4 shows the application installation process that involves access right delegation through capability propagation. Please note that registration process is required by the application developer in order to receive a certificate from a trusted external entity, e.g. BETaaS Apps Store, before the application can be developed, which is a normal procedure in any cloud platform. The application can start installation procedure by including a manifest and its certificate. Application manifest is essentially an XML document that contains a list of required services and other requirements, e.g. QoS, trust score, etc, to run the application properly. After certificate validation by the Security Manager (SecM). It is important to mention that the installation request can arrive at any GW (GW X in Fig. 5.4) in the local cloud and the services

required by the application can be provided by all the other GWs. Therefore, Service Manager (SM) of GW X sends a polling to all participating GWs asking them whether they provide services required by the application. Notice that the illustration in Fig. 5.4 has been really shortened to avoid irrelevant sequence diagram that involves other managers (xxM). Upon receiving a list of equivalent services by all GWs, the SM would then request the ${}_{ext}CAP_n$s of all the equivalent services. Thanks to the access right delegation feature, the SecM of GW X can delegate ${}_{ext}CAP_n$s that correspond to all equivalent services provided by all GWs, thus providing simplicity and flexibility in the application installation. As mentioned earlier in Section 5.4.1, when application accesses the service for the first time, the GW service provider is able to validate the delegated ${}_{ext}CAP_n$ by verifying the GW X's certificate. In the same way, the application can further delegate the ${}_{ext}CAP_n$ to its user (assuming the application is hosted the external server), and the GW service provider is able to validate the delegated ${}_{ext}CAP_n$ by verifying application's certificate. However, the user of that particular application is not able to delegate the ${}_{ext}CAP_n$ since it is registered in the external server, thus the corresponding GW is not able to validate the ${}_{ext}CAP_n$. Nevertheless, if the user is registered in the local cloud platform itself, then the user is able to delegate its ${}_{ext}CAP_n$ any external applications. Thus, it provides flexible yet restricted access control by secure certificate or user credential verification.

5.5 Conclusion

Despite having a critical role, access control in the M2M local cloud platform is faced with unique challenges due to distributed nature of the platform. A capability-based access control coupled with the key management mechanism have been proposed in this paper, providing a secure and scalable access control mechanism with flexible right delegation feature. Some use cases have been presented to show how the proposed approach can provide the desired features in some operations of the real platform. Although the use cases are taken from BETaaS project, it is also applicable to any cloud platform.

Currently, the proposed access control mechanism is still under development in the context of BETaaS project. It is therefore in our interest to test how the proposed capability-based access control approach will work in the real platform deployment and what results can be obtained out of it.

5.6 References

[1] Bayu Anggorojati, Parikshit N. Mahalle, Neeli R. Prasad, and Ramjee Prasad. Secure access control and authority delegation based on capability and context awareness for federated iot. In Fabrice Theoleyre and Ai-Chun Pang, editors, *Internet of Things and M2M Communications*. River Publisher, 2013.

[2] BETaaS. D3.1.2 - betaas architecture. Technical report, Building the Environment for the Things as a Service (BETaaS), 2014.

[3] L. Gong. A secure identity-based capability system. In *Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on*, pages 56 –63, may 1989.

[4] S. Gusmeroli, S. Piccione, and D. Rotondi. A capability-based security approach to manage access control in the internet of things. *Mathematical and Computer Modelling*, 58(5-6):1189 – 1205, 2013.

[5] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[6] George Labropoulos. D1.2.1 - user and system requirements. Technical report, BETaaS, 2012.

[7] OAuth. Oauth 2.0. *http://oauth.net*.

[8] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266 – 2279, 2013.

[9] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *Computer*, 29(2):38 –47, feb 1996.

[10] Xively. Api docs - security.

6

Intrusion Detection Game in Access Control System for the M2M Local Cloud Platform: A Game Theoretical Approach

A distributed M2M local cloud platform which consists of distributed M2M gateways, needs to be equipped with an Intrusion Detection System (IDS) to monitor its resources against security attacks, especially from the insider, e.g. another gateway within the local cloud. In this chapter, the interaction between rational attacker and defender in the context of an M2M local cloud platform as a multi-stage Bayesian game is studied. In this game formulation, a defender is able to update its belief upon the maliciousness of the attacker. The feasible Nash equilibrium of the game is reviewed and an analytical framework for the rational attacker and defender is provided for a given set of resources with different security values under some constraints on the attack and monitor resources. In the numerical analysis, it can be shown that by having multiple resources to be attacked and/or monitored simultaneously provides a kind of diversity which helps to improve the belief update of the defender.

6.1 Introduction

The attacks coming from insider is a great security threat to the IT systems and networks in general, not only in the impact but also in the quantity. To detect and mitigate such threat is a difficult task for the access control system which requires to control access permission to important resources within the system through a set of policies [5]. To this end, an IDS is needed as the integral part of the access control system to detect the various threats.

A distributed local cloud concept in the M2M system deployment has been proposed recently in BETaaS project [9]. The local cloud platform is hosted by multiple M2M gateways cooperating with each other in order to leverage the set of M2M services provided by each gateway. The M2M gateway can be a router, set-top box, or other computing devices (e.g. even tablet and smart phone) that act as a gateway to the M2M devices, thus a local cloud is equivalent to a network of distributed M2M gateways.

Having distributed M2M gateways cooperating with each other forming a local cloud, it is necessary for each gateway to define a set of access control policies to restrict access to its own set of resources from other nodes within a local cloud, including the malicious ones. Here, resources of a gateway include sensor, actuator, data, and service, with different levels of security values, i.e. depending on their cost and importance. The node refers to the M2M gateway, or gateway in short, because gateway is the main actor that enables local cloud, and malicious gateway is a gateway that would cause security threat to the network by attacking the resources owned by another gateway.

We can imagine a situation where a malicious node disguises as a good node such that it can join the local cloud, but once it becomes part of the cloud it would cause a huge damage to the system. For example it could manipulate access right of an actuator controlled by a gateway, e.g. to open a gate or turning on or off some switches, stealing some sensitive data from sensors, and so on. We should keep in mind that this local cloud environment is heterogeneous and the human interaction should be as minimum as possible. Therefore, the IDS should be able to learn and update its knowledge based on the interaction with the other nodes.

This chapter models and analyses the interaction between malicious node (attacker) and the regular node with IDS (defender) with game theory, in order to suggest the best strategies for both sides. Game theory is an analytical framework used to model rational decision making or strategies of multiple agents having different objectives interacting with each other in the same system. Numerous works related to game theory and IDS have been carried out,

each with different objectives [1, 11, 4, 3]. In this chapter, the game is formulated as a multi-stage Bayesian game, as the defender has no prior knowledge about the node it is interacting with, i.e. whether it is malicious or innocent. Through the interactions, which is modelled in a multi-stage game, the defender is able to update its belief about the attacker intention via Bayes rule. On a side note, further action can be taken by the defender, e.g. removing the malicious gateway from the local cloud, once its belief about the attacker being malicious converges to one. Finally, an optimum strategy for both attacker and defender will be derived by considering their respective costs and benefits.

The remaining of this chapter is organized as follows: A review of some related works within the field of game theory in security and intrussion detection is described in section 6.2. A brief overview of a M2M local cloud system along with its security consideration is presented in section 6.3. The game model and definition of sensible resource set are described in section 6.4. The analysis of proposed multi-stage Bayesian game is explained section 6.5. A numerical analysis is performed in section 6.6. Finally, the conclusion and future outlooks are given in section 6.7.

6.2 Related works

Game theory has been applied widely in modelling the network security system, including the interaction between IDSs and the attackers. Alpcan and Başar [1] model the intrusion detection in access control system as a non-cooperative non-zero-sum game in both finite and continuous-kernel versions. The IDS sensor network is also introduced in their model as a fictitious player which models the imperfect detection capabilities of the sensor. The existence and uniqueness of the Nash Equilibrium (NE) was shown, and the repeated games of the continuous-kernel version was studied analytically and numerically by the authors. Bloem *et al.* further developed an algorithm to optimally allocate between system administrator's response under the time and effort constraints in managing the attacks, and the automatic response to the attacks with some imperfections [2], as a discrete model based on the continuous game version in [1]. The overall cost function is minimized as the linear programming optimization problem, and the simulation results of the algorithm are presented by the authors.

Y. Liu *et al.* [11] model the attacker and IDS in the ad hoc wireless networks as a Bayesian game, in both static and dynamic or repeated game. The Perfect Bayesian Equilibrium (PBE) of the game is shown, and a hybrid IDS at the

defender side is proposed in order to balance the trade off between the energy cost and monitoring gain. Chen *et al.* [4] proposed a framework which applies two stages of game theoretical models for economical deployment of intrusion detection agent. In their model, the first scheme models a two-person non-cooperative game between the attacker and IDS agent, and a unique security risk value, which is the NE derived from the proposed pay-off functions, is assigned to the agent. The second scheme uses the security risk value to compute the Shapley value of intrusion detection agent in order to grouped the IDS agents into coalition groups by the various threat levels so as to provide fair and optimal IDS deployment.

Chen and Leneutre [3] model the game theoretical framework for IDS in heterogeneous networks consisting of nodes with different non-correlated security assets. In the model, a non-cooperative game is considered and a set of sensible targets with various attack and monitor resource constraints is defined, in order to analyse and derive the NEs. Various feasible NEs for different types of game, i.e. two-person game, multiple attackers/defenders, Stackelberg Network Intrusion Detection Game, and the game with generalized attack model, are studied, and a numerical study is performed to validate the analytical results.

A dynamic Bayesian game between regular and malicious nodes in mobile ad-hoc networks is studied in [10]. In the model, the regular node would either cooperate or decline the cooperation with another node, and the malicious node would either attack or cooperate with another node in order to confuse or deceive the opponent. On top that, the regular node has an option to report the malicious node at certain point, i.e. after it receives enough evidence provided that the reporting gives better pay-off than the other actions. On the other hand, the malicious node has an option to flee provided that the pay-off of fleeing is higher than the risk of being reported. The pure, mixed, and PBE strategies of both types of players are studied, and the simulation results to evaluate those strategies are presented.

6.3 The M2M local cloud system and security consideration

The M2M local cloud system considered in this chapter refers to the platform being introduced and developed in BETaaS project. According to BETaaS description in [9], it is *a content-centric platform distributed over a local cloud,*

hosted by gateways, providing an environment for applications accessing M2M services and devices through a set of services.

The high level architecture consists of three layers: *Service* layer, *Logical* layer, and *Physical* layer. The *service* layer contains a set of semantic and high level services that are built using the functionality of the TaaS reference model and can be used by the external M2M applications. The *logical* layer consists of a TaaS reference model which contains the core logic of the BETaaS platform that allows the implementation of service modules through a unified representation of an M2M model, and an *adaptation layers* which provides an abstract view of a generic M2M system to the TaaS over various M2M system specific implementation (e.g. ETSI M2M or proprietary systems). The physical layer consists of smart things, i.e. the IoT or M2M devices. Although it is part of the BETaaS architecture, the physical layer is out of the research scope in BETaaS. However, two relevant initiatives in M2M, e.g. ETSI M2M [6] and IoT-A [8], are used as reference to create an appropriate abstraction of TaaS reference model. Figure 6.1 depicts the high level architecture of the M2M local cloud system in BETaaS.

According to this description of the M2M local cloud system, it is obvious that the main functionality of local cloud system resides in the *logical* and *service* layers. On the practical deployment perspective, these layers are implemented in a BETaaS gateway, in which one or a set of distributed BETaaS gateway(s) are able to establish a local cloud. A BETaaS gateway is a logical entity which reside in a physical gateway e.g. router, set-top box, and other computing devices like tablet and smart phone. It is important to note that a physical gateway may contain multiple BETaaS gateways, i.e. by means of virtualization. However, to simplify our modelling and analysis, a BETaaS gateway is considered to be equivalent with a physical gateway, and will be referred only as *gateway* throughout the rest of this chapter.

As mentioned earlier, at this point a local cloud can be seen as a distributed network consisting of multiple gateways, and therefore the security consideration is focused on the gateway level upon its interaction, especially with the other gateways within the local cloud. Given that, having an IDS in a gateway is a realistic assumption as an IDS is usually implemented as an autonomous software agent which resides at network devices, e.g. router, server, or terminal.

Based on the system description and our previous argument of a big chance in having great threat or enemy from within the system, an intrusion detection game between two gateways (attacker and defender) within a local cloud is considered. The defender naturally has no full knowledge of whether or not

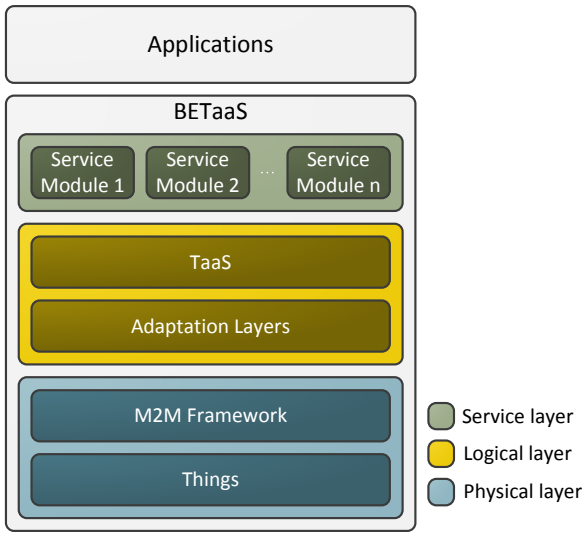

Figure 6.1: High level architecture of the M2M local cloud system based on BETaaS

the attacker, i.e. another gateway, is malicious, and this knowledge could be obtained through some interactions. Therefore, a multi-stage Bayesian game with two-player is considered as this model is suitable to model such interaction. In the rest of this chapter, the dynamic Bayesian game formulation and solutions to model such interaction will be presented.

6.4 Single stage game model

A local cloud system with a fixed number of gateway in the network is considered. It is assumed that each gateway, i.e. acting as a defender, is equipped with an IDS which can detect any attacker in the network or local cloud.

The players participating in the game are a potential attacker and a defender, denoted by index A and D respectively. The defender has a set of resources $R = \{1, 2, ..., N\}$ to be protected. To avoid any confusion with the term resource used on different context in this chapter, this set of resources R will be reffered as *asset* throughout the rest of the chapter. The type of player A is defined in a finite set $\Theta_A = \{\theta_{A0}, \theta_{A1}\}$, where θ_{A0} denotes a regular type and θ_{A1} denotes malicious type. The type of player A is a private information for itself, i.e. whether or not he is a malicious node unknown to the player D. On the other hand, the type of player D is only regular and its type is known

by both players A and D. Given that, player D's belief about the type of player A being malicious is expressed as a prior probability, denoted by μ_0.

The interaction between attacker and defender is modelled as a non-cooperative game, which extends the basic two-player game formulation in [3] into a Bayesian game. The malicious type of player A has two pure strategies $\mathcal{A}_{Am} = \{a_{A0}, a_{A1}\}$, where a_{A0} denotes strategy *Not attack* and a_{A1} denotes strategy *Attack*, and the regular type of player A has only one pure strategy: $\mathcal{A}_{Ar} = \{a_{A0}\}$. Here, the objective of the malicious type of player A is to attack a set of assets R without being detected. On the other hand, the player D has two pure strategies: $\mathcal{A}_D = \{a_{D0}, a_{D1}\}$, where a_{D0} denotes strategy *Not monitor* and a_{D1} denotes strategy *Monitor*. Here, the action *Monitor* refers to monitoring a set of its own assets R so as to minimize its loss or optimize its payoff.

It is assumed that each asset $i \in R$ worth of a security asset value W_i, where $W_i > 0$. In reality, W_i could be an economic value of the asset, whose value may depend on how valuable the information it holds, its role in the network, etc. Therefore $-W_i$ expresses the loss of security value due to a successful attack, e.g. loss of important data or data integrity, loss of revenue due to damage of a certain asset, etc. It is also assumed that upon a successful attack of asset i, both player A and D receive an equal gain/loss W_i, i.e. the payoff for player A and D is W_i and $-W_i$ respectively. In this chapter, it is assumed that the security assets of different assets are independent. This assumption holds for ad-hoc network scenario where each node operates independently of each other [3], e.g. in our case ad-hoc nodes represent the the sensors or actuators operating "behind" the gateway.

In general, the player A's strategy is to attack a set of assets R under a probability distribution $\mathbf{p} = \{p_1, p_2, ..., p_N\}$, where p_i is the probability of attacking asset i and $\sum_{i \in R} p_i \leq P \leq 1$ indicates the resource constraint of the player A. On the defender D side, its strategy is to monitor a set of assets R in order to detect the attacks with a probability distribution $\mathbf{q} = \{q_1, q_2, ..., q_N\}$, where q_i is the probability of monitoring the asset i and $\sum_{i \in R} q_i \leq Q \leq 1$ indicates the player D's resource constraint.

Table 6.1 illustrates the payoff matrix of the attacker/defender Bayesian game for the asset i in strategic form. In the matrix, the x and y denote the detection rate (true positive) and false alarm rate, i.e. false positive rate, of the IDS respectively, and $x, y \in [0, 1]$. The cost of attacking and monitoring of asset $i \in R$ are denoted by $C_a W_i$ and $C_m W_i$ respectively, and are assumed to be proportional to the security asset value of i, W_i. $C_f W_i$ denotes the cost of a false alarm. In this study, the $C_a, C_m \in [0, 1)$ is assumed so as to give incentive to player A/D to attack/monitor.

Table 6.1: Strategic form of the attacker/defender Bayesian game for the asset i

	Monitor	Not monitor
Attack	$(1-2x)W_i - C_aW_i$, $(2x-1)W_i - C_mW_i$	$W_i - C_aW_i$, $-W_i$
Not attack	0, $-yC_fW_i - C_mW_i$	$0, 0$

(a) Player A is malicious with prior probability μ_0

	Monitor	Not monitor
Not attack	0, $-yC_fW_i - C_mW_i$	$0, 0$

(b) Player A is regular with prior probability $1 - \mu_0$

In Table 6.1a, for the strategy combination (Attack, Not monitor), player D payoff is $-W_i$, and the malicious type of player A's payoff is his gain of success minus the attacking cost, i.e. $W_i - C_aW_i$. For the strategy combination (Attack, Monitor), the overall payoffs of both players are defined as the expected gain/loss of detecting the attack minus the cost of monitoring/attacking asset i, i.e. C_mW_i and/or C_aW_i respecively. The expected gain of detecting the attack depends on the value of x, which is $xW_i - (1-x)W_i = (2x-1)W_i$, where $1-x$ is the false negative rate. On the other hand, the player A would receive an equivalent gain due to the defender's loss, i.e. $(1-2x)W_i$. For the other two strategy combinations, when player A plays *Not attack*, his payoff is always 0. On the other hand, player D's payoff is 0 if she does not monitor, and a monitoring cost of C_mW_i plus an expected loss of yC_fW_i due to false alarms would be utilized if the monitor strategy is chosen. Please note that for the payoff matrix in Table 6.1b, i.e. if player A is a regular type, it basically has a similar payoffs combination to the second row of the payoff matrix in Table 6.1a.

The overall payoffs of both player A and D are defined as the following utility functions:

$$\begin{aligned} U_A(\mathbf{p}, \mathbf{q}) &= \sum_{i \in R} [p_i q_i W_i (1 - 2x - C_a) + (1 - q_i) W_i (1 - C_a)] \\ &= \sum_{i \in R} p_i W_i (1 - 2x q_i - C_a) \end{aligned} \tag{6.1}$$

$$\begin{aligned} U_D(\mathbf{p},\mathbf{q}) = &\sum_{i\in R}[\mu_0(p_i q_i W_i(2x-1-C_m) + (1-p_i)q_i W_i(-yC_f - C_m) \\ &+ p_i(1-q_i)(-W_i)) + (1-\mu_0)q_i W_i(-yC_f - C_m)] \\ = &\sum_{i\in R} q_i W_i(p(2x+yC_f)\mu_0 - (yC_f + C_m)) - \sum_{i\in R}\mu_0 p_i W_i \end{aligned} \tag{6.2}$$

6.4.1 Sensible set of assets

In order to optimize its strategy under some attack resource constraint, a rational attacker A would rather to focus its attacks to some assets to minimize the risk of being detected, which has been shown by Chen and Leneutre in [3]. Given a set of resource R, it is assumed that each asset has a security asset value W_i sorted in the following order: $W_1 \geq W_2 \geq \cdots \geq W_N$, and the sensible assets set R_S and the quasi-sensible asset set R_Q are defined such that

$$\begin{cases} W_i > \frac{|R_s|.(1-C_a)-2xQ}{(1-C_a)\left(\sum_{j\in R_s}\frac{1}{W_j}\right)}, \forall i \in R_S \\ W_i = \frac{|R_s|.(1-C_a)-2xQ}{(1-C_a)\left(\sum_{j\in R_s}\frac{1}{W_j}\right)}, \forall i \in R_Q \\ W_i < \frac{|R_s|.(1-C_a)-2xQ}{(1-C_a)\left(\sum_{j\in R_s}\frac{1}{W_j}\right)}, \forall i \in R - R_S - R_Q \end{cases} \tag{6.3}$$

where $|R_S|$ is the cardinality of R_S, and $R - R_S - R_Q$ denotes the set of assets in set of assets R but neither in R_S nor R_Q. According to [3], R_S consists of N_A assets with the largest security asset values such that they incentive for player A to attack them. On the other hand, attacking any asset $i \in R - R_S - R_Q$ gives no incentives to attacker A. In other words, focusing only to attack the assets set R_S and R_Q is enough to maximize the attacker's payoff, while the security assets values of set of assets $R - R_S - R_Q$ are not "attractive" enough to draw attacker's A attention, even if they are not monitored by the defender D. Consequently, the defender D has no incentive to monitor any asset $i \in R - R_S - R_Q$.

6.5 Proposed multi-stage Bayesian game

To analyse the multi-stage Bayesian intrusion detection game, it is assumed that the single stage Bayesian game model is played repeatedly in each time period t_k, $k = 0, 1, ...$ The payoffs of each player in each stage of the game

are the same as the payoffs defined in Section 6.4. To simplify the analysis, the discount factor with respect to each player's payoff is not considered, thus the payoffs remain the same in each stage. Another assumption is that each of the gateway has been authenticated upon joining the local cloud, thus the identity of all the gateway, i.e. player, is maintained throughout the game.

At the beginning of each stage game, e.g. time period t_k, each player chooses its action among its own pure strategy set simultaneously that is revealed at the end of the stage. In this case, a malicious type of player A will choose its action at time t_k, $a_A(t_k)$, from its strategies set $\mathcal{A}_{Am} = \{a_{A0}, a_{A1}\}$, and similarly player D chooses action $a_D(t_k)$ from the strategies set $\mathcal{A}_D = \{a_{D0}, a_{D1}\}$. In addition, the single stage game model in section 6.4 implicitly suggests that the game is played by each player using *mixed strategy*, i.e. a probability distribution over pure strategies [7]. Similarly in the multi-stage Bayesian game, a strategy of each player prescribes a behaviour strategy which essentially is a probability distribution σ that maps the set of possible set of action history profile $h(t_k) = (a(t_0), a(t_1), ..., a(t_{k-1}))$ and types θ into the action spaces a, i.e. $\sigma(a|\theta, h(t_k))$ [7]. In our case, the behaviour strategy of player A is defined as $\sigma_A(a^i_A(t_k)|\theta_A, h^i_A(t_k))$, where θ_A denotes any type of player A within a set Θ_A, $h^i_A(t_k) = (a^i_A(t_0), a^i_A(t_1), ..., a^i_A(t_{k-1}))$ indicates the history profile of player A with respect to asset $i \in R$ that belongs to player D at the beginning of stage t_k, and $a^i_A(t_k)$ represents the action profile of player A with respect to asset $i \in R$ at stage t_k. Likewise, the behaviour strategy of player D is defined as $\sigma_D(a^i_D(t_k)|\theta_D, h^A_i(t_k))$.

Following the mixed strategies of both players defined in section 6.4, i.e. the attack probability $\mathbf{p}$ and monitor probability $\mathbf{q}$ over the asset $i \in R$, the set of behaviour strategy σ of each player in each stage game t_k can be defined as follows:

$$\sigma_A(a^i_{A1}(t_k)|\theta_A, h^i_A(t_k)) = p_i$$

$$\sigma_A(a^i_{A0}(t_k)|\theta_A, h^i_A(t_k)) = P - p_i$$

$$\sigma_D(a^i_{D1}(t_k)|\theta_D, h^i_D(t_k)) = q_i$$

$$\sigma_D(a^i_{D0}(t_k)|\theta_D, h^i_D(t_k)) = Q - q_i$$

Player D who has no knowledge of player A's type θ_A, has to choose its strategy at each stage game t_k according to its belief about the type of player A at the beginning of the stage. By observing the action and history profile of player A towards the asset $i \in R$ at the end of the stage game, player D is able to update its belief on the posterior probability distribution

$\mu_D(\theta_A|a^i_A(t_k), h^i_A(t_k))$ that will be used at the beginning of the subsequent stage game.

The posterior belief update of player D about the maliciousness of player A based on its observation of player A's action and history profile at stage game t_k to the stage t_{k+1} is done based on Bayes rule, and is computed as follows:

$$\mu_D(\theta_{A1}|a^i_A(t_k), h^i_A(t_k)) = \frac{\mu_D(\theta_{A1}|h^i_A(t_k))P(a^i_A(t_k)|\theta_{A1}, h^i_A(t_k))}{\sum_{\theta_A \in \Theta_A} \mu_D(\theta_A|h^i_A(t_k))P(a^i_A(t_k)|\theta_A, h^i_A(t_k))} \quad (6.4)$$

where $\mu_D(\theta_A|h^i_A(t_k)), h^i_A(t_k) > 0$, $P(a^i_A(t_k)|\theta_{A1}, h^i_A(t_k))$ is the probability that any action a_A of player A is observed at stage t_k, given the type of the player A being malicious and the history of the set of action from player A throughout the game, and $\mu_D(\theta_{A1}|h^i_A(t_k))$ is the prior belief of player D at stage t_k which was the result of the posterior belief update at stage t_{k-1}.

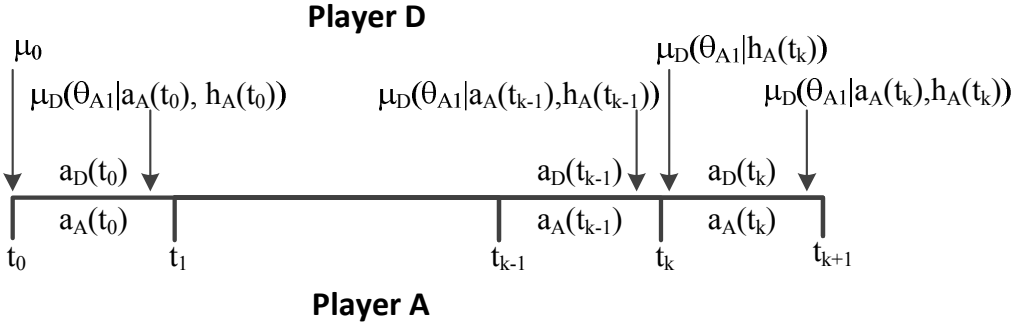

Figure 6.2: Illustration of the player D's posterior belief update in the multi-stage Bayesian game

Figure 6.2 illustrates the player D's posterior belief update process using Bayes rule in the multi-stage game between player A and D, to give a better understanding of the multi-stage game.

To compute posterior belief according to Equation 6.4, all possible conditional probabilities $P(a^i_A(t_k)|\theta_A, h^i_A(t_k))$ need to be specifically defined. In our case, this depends on the probability of the player A launching an attack p, the detection rate and false positive rate of the IDS, x and y respectively. Hence, the conditional probabilities can be computed as follows:

$$P(a_{A1}(t_k)|\theta_{A1}, h^i_A(t_k)) = xp_i + y(P - p_i)$$
$$P(a_{A0}(t_k)|\theta_{A1}, h^i_A(t_k)) = (1 - x)p_i + (1 - y)(P - p_i)$$
$$P(a_{A1}(t_k)|\theta_{A0}, h^i_A(t_k)) = y$$
$$P(a_{A0}(t_k)|\theta_{A0}, h^i_A(t_k)) = 1 - y$$

where $1-x$ indicates the false negative (miss detection) rate, and $1-y$ indicates the true negative rate.

6.5.1 PBE analysis

In a multi-stage game, the best strategy of each player should represent a Nash Equilibrium in every stage game of the whole game which known as subgame perfect equilibrium. Those strategies are determined by considering the responses or reactions of the other players in each stage of the game. However, the opponent's in a dynamic game with incomplete information responses are highly dependent on the player's belief about the opponent. To extend the subgame perfection in a dynamic Bayesian game, i.e. PBE, it is necessary to obtain strategies that yield a Bayesian Nash equilibrium not only for the whole game, but also at every stage of the game. In order to satisfy this, the players' belief about the opponent must be specified at the beginning of each stage game. As mentioned earlier, the player's belief is updated according to the Bayes rule.

In what follows, the existence of PBE in the proposed multi-stage IDS game is shown and then the mixed strategies equilibrium are determined. First, it is shown that the proposed multi-stage game satisfies the Bayesian conditions B(i)-B(iv) and equilibrium condition P, which guarantee that the incomplete-information game has a PBE [7].

Lemma 1. *The described multi-stage IDS game satisfies the four Bayesian conditions [7]:*

B(i) Posterior beliefs are independent, and all types of player D *have the same beliefs, for all* θ*,* t*, and* $h(t_k)$*. It requires that even unexpected events will not change the independence assumption for the type of the opponents.*

B(ii) Bayes' rule is used to update beliefs from $\mu_D(\theta_A|h^i_A(t_k))$ *to* $\mu_D(\theta_A|h^i_A(t_{k+1}))$ *whenever possible.*

B(iii) The players do not signal what they do not know.

B(iv) All players must have the same belief about the type of another player.

Proof. B(i) is satisfied because player D has only one type. From the proposed belief updating system, the game satisfies B(ii). Condition B(iii) means $\mu_D(\theta_{A1}|a^i_A(t_k), h^i_A(t_k))$; $\mu_D(\theta_{A1}|\hat{a}^i_A(t_k), h^i_A(t_k))$, if $a^i_A(t_k) = \hat{a}^i_A(t_k)$. In our IDS game, the belief update of player D about player A's type is only influenced by player A's signal, thus B(iii) is satisfied. Since there are only two players

are in the game at any stage game, and there are no other players influencing the belief updates of the two players, hence it satisfies condition B(iv). □

The main point stated in Lemma 1 is that each player's belief updating is consistent in every stage game. By assuming that the players are rational, player D's optimal strategy at each stage game is to maximize its payoff according to his new beliefs.

Lemma 2. *The described multi-stage attacker/defender game satisfies the equilibrium condition P for multi-stage games of incomplete-information:*

(P) For each player n, type θ_n, player n's alternative strategy σ'_x, and history $h(t_k)$, the expected payoff achieved by employing strategy σ_n, denoted by u_n, satisfies the following condition:

$$u_n(\sigma|h(tk), \theta_n, \mu(\cdot|h(tk))) \geq u_A((\sigma'_n, \sigma_{-n})|h(tk), \theta_n, \mu(\cdot|h(tk)))$$

In principal, condition P states that each player's behaviour strategy is sequential rational in each stage game. To proof Lemma 2, the optimum behaviour strategy of each player will be shown.

Proof. Player D's optimal behaviour strategy σ^*_D with respect to his beliefs about player A's type $\mu_D(\theta_A|a^i_A(t_k), h^i_A(t_k))$ at stage game t_k satisfies the following relation:

$$u_D((\sigma_A, \sigma^*_D)|\theta_D, h^i_A(tk), \mu_D(\cdot)) \geq u_D((\sigma_A, \sigma'_D)|\theta_D, h^i_A(tk), \mu_D(\cdot)) \tag{6.5}$$

where σ'_D is an alternative behaviour strategy of defender D, and $h^i_A(tk)$ is the action history profile of player A with respect to the asset $i \in R$, $\mu_D(\cdot)$ is the abbreviation of $\mu_D(\theta_A|a^i_A(t_k), h^i_A(t_k))$, and $u_D(\cdot)$ is the expected payoff of defender D under strategy profile (σ_A, σ^*_D) at stage game t_k.

In the same manner, attacker A's optimal behaviour strategy σ^*_A with respect to his beliefs $\mu_A(\theta_D|a^i_D(t_k), h^i_D(t_k))$ at stage game t_k satisfies the following relation:

$$u_A((\sigma^*_A, \sigma_D)|\theta_A, h^i_D(tk), \mu_A(\cdot)) \geq u_A((\sigma'_A, \sigma_D)|\theta_A, h^i_D(tk), \mu_A(\cdot)) \tag{6.6}$$

where σ'_A is an alternative behaviour strategy of player A, and $h^i_D(tk)$ is the action history profile of player D with respect to the asset $i \in R$, $\mu_A(\cdot)$ is the abbreviation of $\mu_A(\theta_D|a^i_D(t_k), h^i_D(t_k))$, and $u_A(\cdot)$ is the expected payoff of

player A under strategy profile (σ_A^*, σ_D) at stage game t_k. Since player D has only one type, Equation 6.6 reduces to

$$u_A((\sigma_A^*, \sigma_D)|\theta_A, h_D^i(tk)) \geq u_A((\sigma'_A, \sigma_D)|\theta_A, h_D^i(tk)) \tag{6.7}$$

Based on the optimum strategies of each player defined in Equation 6.5 and 6.7, given player D's belief μ_D, the multi-stage attacker/defender game has a strategy pair $\sigma = (\sigma_A^*, \sigma_D^*)$ that satisfies the above inequality formula, hence the equilibrium condition P is satisfied. □

Theorem 1. *The described multi-stage IDS game has a perfect Bayesian equilibrium.*

Proof. Since the described multi-stage attacker/defender game satisfies the four Bayesian conditions B(i)-B(iv) (Lemma 1) and the equilibrium condition P (Lemma 2), the game has a strategy profile (σ, μ), where $\sigma = (\sigma_A^*, \sigma_D^*)$ is a strategy pair for the two players, and $\mu = (\mu_A(\theta_D|h_D^i(t_k)), \mu_D(\theta_A|h_A^i(t_k)))$ is the vector of beliefs for the two players. Note that μ_A is not needed since player D's type is common knowledge. By the definition of PBE [7], (σ, μ) is a PBE. □

Next, the types of potential PBE, namely *separating equilibrium*, *pooling equilibrium*, and *hybrid equilibrium* [7], are analysed. In our IDS game, the *separating equilibrium* cannot be satisfied because the *Attack* launched by the malicious type of player A cannot perfectly reveals its type to player D due to the error detection rate that the IDS has. On the other hand, player A only wants to play *Not attack* if by playing the opposite it will receive a negative reward. Therefore, it can be seen that the only type of PBE exists in our game is of *hybrid* or *semi-separating equilibrium*, in which both players may randomize or mix their strategies in order to maximize their respective payoff, while none of them have incentive to deviate from the equilibrium strategies. This also confirms the existence of mixed strategy equilibrium solution in each stage game.

Please recall that there is a set of assets $i \in R$ with different security values to be attacked and monitored by player A and D respectively, under a given resource constraint to each player. In such a case, the mixed strategy equilibrium should be determined by analyzing different cases, i.e. depending on the resource constrains. For this purpose, a similar analysis explained in [3] will be followed.

First, let $(\mathbf{p}^*, \mathbf{q}^*, \mu(\cdot))$ denote the PBE of the game, and given the utility functions of each player, U_A and U_D, the following relations are satisfied:

$$\begin{aligned} 0 \leq & (1 - 2xq_i^* - C_a)W_i = (1 - 2xq_j^* - C_a)W_j \\ \geq & (1 - 2xq_l^* - C_a)W_l \forall i, j, l \in R, p_i^*, p_j^* > 0, p_l^* = 0 \end{aligned} \tag{6.8}$$

$$\begin{aligned} 0 \leq & [p_i^*(2x + yC_f)\mu_D(\theta_{A1}|\cdot) - (yC_f + C_m)]W_i \\ = & [p_j^*(2x + yC_f)\mu_D(\theta_{A1}|\cdot) - (yC_f + C_m)]W_j \\ \geq & [p_l^*(2x + yC_f)\mu_D(\theta_{A1}|\cdot) - (yC_f + C_m)]W_l \\ & \forall i, j, l \in R, q_i^*, q_j^* > 0, q_l^* = 0 \end{aligned} \tag{6.9}$$

The relationship shown in Equation 6.8 can be explained as follows: if $(1 - 2xq_i^* - C_a)W_i < 0$, player A would better choose $p_i^* = 0$; if $(1 - 2xq_i^* - C_a)W_i < (1 - 2xq_j^* - C_a)W_j$, then player A has incentive to decrease p_i^* and increase p_j^*; if $(1 - 2xq_i^* - C_a)W_i < (1 - 2xq_l^* - C_a)W_l$, then player A will get more payoff by adding p_i^* to p_l^* and setting $p_i^* = 0$. Similarly, Equation 6.9 can also be explained in the same way.

Based on resource constraint of each player, the following cases are considered:

1. $\sum_{i \in R} q_i^* = Q$ and $\sum_{i \in R} p_i^* = P$
2. $\sum_{i \in R} q_i^* < Q$ and $\sum_{i \in R} p_i^* = P$
3. $\sum_{i \in R} q_i^* < Q$ and $\sum_{i \in R} p_i^* < P$

In *Case 1*, both players use up all their resources to attack or monitor. It happens when the security assets W_i from the set of resources are considerably high as compared to the attacking or monitoring cost, therefore both players are trying to attack and monitor the assets that have the biggest security values (as pointed out in Section 6.4.1) using all of their attacking/monitoring assets. In *Case 2*, player A uses all its attacking resource to the limit, while it does not give incentive for player D to utilize all its monitoring resources, due to high monitoring cost as compared to the gain it would achieve, i.e. the security values of protected assets. In *Case 3*, both players do not use all of their attacking/monitoring resources because by doing so it does not give so much incentive for both of them. This is due to high attacking and monitoring cost as compared to the security values of the assets.

From the formulation of our multi-stage Bayesian game and the *Theorem 2* in [3], it can be shown that the equilibrium strategies of each player in each

of the case mentioned earlier at stage game t_k, i.e. $p_i^*(t_k)$ and $q_i^*(t_k)$, result in the following strategies:

1) If $N_D \geq N_A$ and $N_A(1-C_a) > 2xQ$, then

$$p_i^*(t_k) \begin{cases} = \frac{P_A}{W_i \sum_{j=1}^{NA} \frac{1}{W_j}} + \frac{yC_f+C_m}{(2x+yC_f)\mu_D(\theta_{A1}|\cdot)} \left(1 - \frac{N_A}{W_i \sum_{j=1}^{NA} \frac{1}{W_j}}\right) & , i \in R_S \\ \in \left[0, \frac{P_A}{W_i \sum_{j=1}^{NA}} + \frac{yC_f+C_m}{(2x+yC_f)\mu_D(\theta_{A1}|\cdot)} \left(1 - \frac{N_A}{W_i \sum_{j=1}^{NA} \frac{1}{W_j}}\right)\right] & , i \in R_Q \\ = 0 & , i \in R - R_S - R_Q \end{cases}$$

$$q_i^*(t_k) = \begin{cases} \frac{1}{2x}\left(1 - C_a - \frac{NA(1-C_a)-2xQ}{W_i \sum_{j=1}^{NA} \frac{1}{W_j}}\right) & , i \in R_S \\ 0 & , i \in R - R_S \end{cases}$$

2) If $N_D \leq N_A$

$$p_i^*(t_k) \begin{cases} = \frac{yC_f+C_m}{(2x+yC_f)\mu_D(\theta_{A1}|\cdot)} & , W_i > W_{N_D+1} \\ \in \left[0, \frac{yC_f+C_m}{(2x+yC_f)\mu_D(\theta_{A1}|\cdot)}\right] & , W_i = W_{N_D+1} \\ = 0 & , W_i < W_{N_D+1} \end{cases}$$

$$q_i^*(t_k) = \begin{cases} \frac{1-C_a}{2x}\left(1 - \frac{W_{N_D+1}}{W_i}\right) & , W_i > W_{N_D+1} \\ 0 & , W_i \leq W_{N_D+1} \end{cases}$$

3) If $N_D \geq N_A$ and $N_A(1-C_a) \geq 2xQ$, then

$$p_i^*(t_k) = \frac{yC_f + C_m}{(2x+yC_f)\mu_D(\theta_{A1}|\cdot)}, q_i^*(t_k) = \frac{1-C_a}{2x} \quad , i \in R$$

in this case $N_D = N_A = N$.

where N_D is the number of security assets that are worth to be monitored by player D and is defined as $N_D = \lfloor (2x + yC_f)P/(yC_f + C_m) \rfloor$.

6.6 Numerical analysis

In this section, some numerical analysis are performed to illustrate our analytical model and the equilibrium in the described multi-stage Bayesian game.

A scenario where the attacking and monitoring costs are relatively high, i.e. due to energy or other constraints, while security values of the assets $W_i, i \in R$,

are not as much high a compared to the attacking and monitoring costs, is considered. In this case, the following cost parameters are set: $C_a = C_m = 0.1$, and $C_f = 0.3$. The detection rate and false positive rate of the IDS are $x = 83.33\%$ and $y = 0.029\%$ respectively. Ten protected assets with normalized security values are considered: $W_i = 0.1 * (11 - i)$, $i \in [1, 10]$. Both attacking and monitoring resources, P and Q, are set to 1.

Table 6.2 shows the mixed strategies equilibrium in each stage game $(\mathbf{p}^*, \mathbf{q}^*)$. Given its attack resource constraint, it can be seen that player A focuses only to attack six most valuable assets out of ten with some probabilities. Consequently, player D has incentive to monitor the same six assets as well.

For the purpose of illustrating how the posterior belief of player D upon observing the player A's action, i.e. $\mu_D(\theta_{A1}|\cdot)$, and how player A's strategies distribution adapts the $\mu_D(\theta_{A1}|\cdot)$ in the multi-stage Bayesian game, a hundred stages game is run in which each player choose its own strategy simultaneously based on the observation in the previous stage. At the beginning of the stage, both $\mu_D(\theta_{A1}|\cdot)$ and $\mu_D(\theta_{A0}|\cdot)$ equal to 0.5. Player A decides to *Attack* or *Not attack* randomly according to the calculated equilibrium strategy $\mathbf{p}^*$, and the same way applies to player D.

Figure 6.3 and 6.4 illustrate how the posterior belief of player D evolves upon its observation to player A's action on the asset $i = 1$ and $i = 5$ respectively, over the game duration. In both figures, observed action 0 means that player D observes *No attack*, and 1 means *Attack* action was observed. It is clear that the posterior belief quickly converges to 1 after observing *Attack* due to high detection rate and low false alarm rate. Furthermore, it is very difficult to reduce the posterior belief to a lower level in the next stage even a

Table 6.2: Mixed equilibrium strategies

i	p_i^*	q_i^*
1	$0.1183 - 0.0176/\mu_D(\theta_{A1}\|\cdot)$	0.2751
2	$0.1314 - 0.0128/\mu_D(\theta_{A1}\|\cdot)$	0.2457
3	$0.1478 - 0.0068/\mu_D(\theta_{A1}\|\cdot)$	0.2089
4	$0.1689 + 0.0008/\mu_D(\theta_{A1}\|\cdot)$	0.1616
5	$0.1971 + 0.0110/\mu_D(\theta_{A1}\|\cdot)$	0.0985
6	$0.2365 + 0.0253/\mu_D(\theta_{A1}\|\cdot)$	0.0102
7	0	0
8	0	0
9	0	0
10	0	0

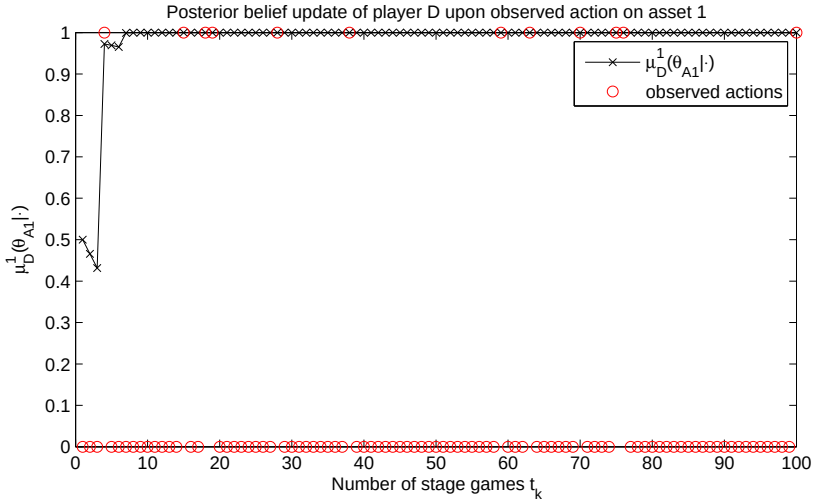

Figure 6.3: Player D's posterior belief update $\mu_D(\theta_{A1}|\cdot)$ upon its observation to player A's action on the asset $i = 1$

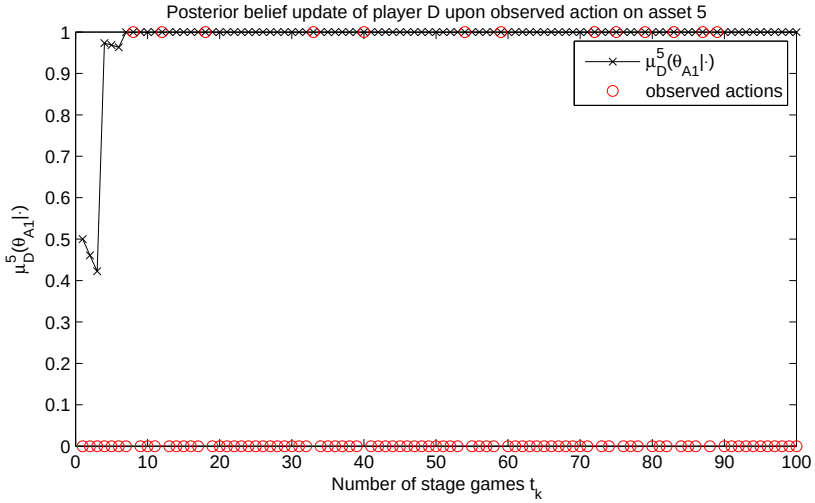

Figure 6.4: Player D's posterior belief update $\mu_D(\theta_{A1}|\cdot)$ upon its observation to player A's action on the asset $i = 5$

No attack action is observed once it is converged to 1. It can also be noticed that having multiple assets to be monitored simultaneously quickly improves the posterior belief update, as chances of observing attacks toward any of the available asset is more than just monitoring a single asset.

Figure 6.5 illustrates how the mixed equilibrium strategy of player A, $\mathbf{p}^*$, over the most valuable assets evolves in the multi-stage Bayesian game. It is

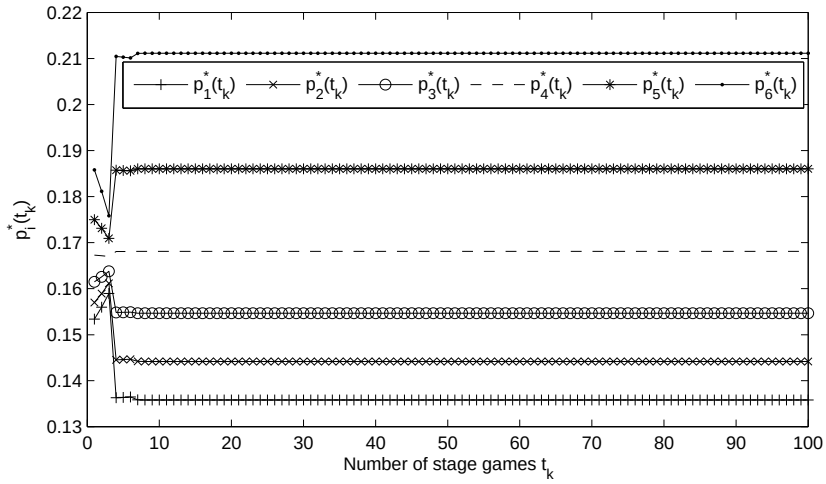

Figure 6.5: Player A's best strategy p^* in each stage game over the most valuable assets

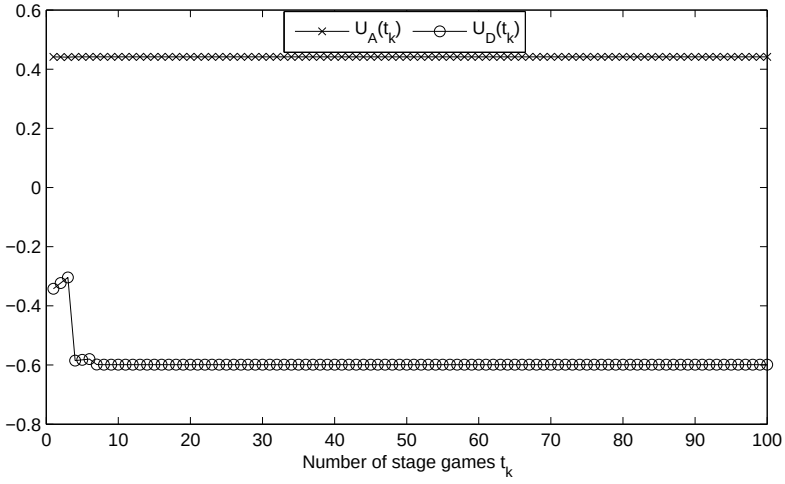

Figure 6.6: Players A's and D's overall payoffs in each stage game

can be seen that at the initial stage of the game until an *Attack* is observed by player D, player A tries to increase or decrease its probability of attacking over different security assets in order to maintain its optimum payoff. However, when an *Attack* is observed, player A pulls back its attack probability to reduce its probability of being caught by player D, thus maintaining its optimum overall payoff. It is obvious as well that $\mathbf{p}^*$ converges to a steady state value once the $\mu_D(\theta_{A1}|\cdot)$ converges to 1.

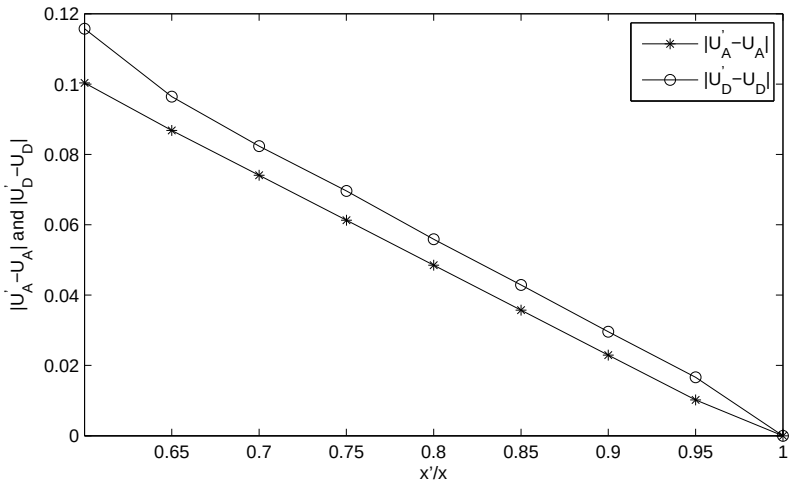

Figure 6.7: Players A's and D's average utilities deviations, i.e. $\left|U_A^{'} - U_A\right|$ and $\left|U_D^{'} - U_D\right|$ vs $\frac{x'}{x}$, from the variation of detection rate, with $x = 83.33\%$ and $y = 0.29\%$

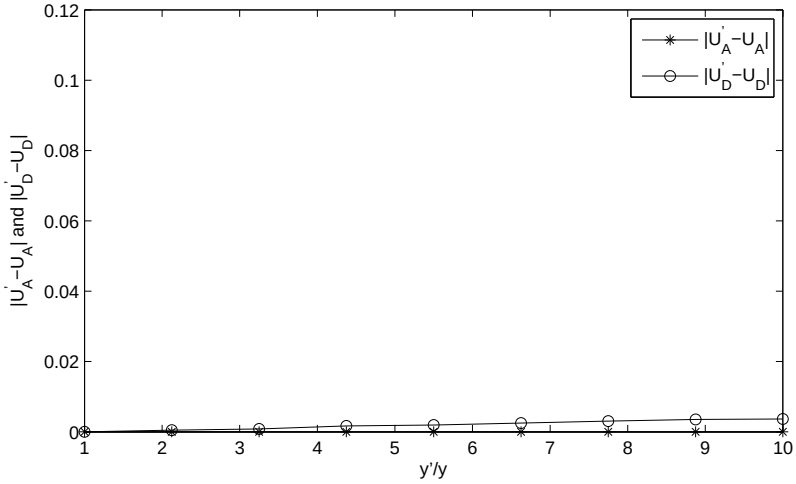

Figure 6.8: Players A's and D's average utilities deviations, i.e. $\left|U_A^{'} - U_A\right|$ and $\left|U_D^{'} - U_D\right|$ vs $\frac{y'}{y}$, from the variation of false alarm rate, with $x = 83.33\%$ and $y = 0.29\%$

Figure 6.6 shows the players A's and D's payoffs in each stage of the game,i.e. $U_A(t_k)$ and $U_D(t_k)$ respectively. It is observed that $U_A(t_k)$ remains constant because it is not a function of $\mu_D(\theta_{A1}|\cdot)$. On the other hand, $U_D(t_k)$

varies as $\mu_D(\theta_{A1}|\cdot)$ varies, and $U_D(t_k)$ reaches a steady state once $\mu_D(\theta_{A1}|\cdot)$ converges to 1.

As the utility functions of both players A and D are influenced by the IDS's performance represented by the detection rate and false positive rate, e.g. x and y respectively, the next question would be how these parameters influence the overall utilities of the players throughout the game. Figure 6.7 and 6.8 show the variations of both player A's and D's average payoffs throughout the game, represented by $\left|U_A^{'} - U_A\right|$ and $\left|U_D^{'} - U_D\right|$ respectively, due to the deviations of IDS's performance parameters, with the default parameters $x = 83.33\%$ and $y = 0.29\%$. In both observations, a series of one hundred stages game is played for 100 times, and then the average utilities in each series are taken before averaging over 100 times of playing. In Figure 6.7, the players' utilities deviations are observed by lowering the detection rate to as low as 0.6 of the default detection rate parameter, i.e. $x' = 50\%$, while keeping the default false alarm rate $y = 0.29\%$. Please note that in this case, the $U_A^{'}$ increases as the x' decreases while the $U_D^{'}$ goes lower as the x' decreases. This behaviour is expected because when the IDS's is lower, player A receives more payoff, while the opposite behaviour applies to player D. In Figure 6.8, the players' utilities deviations are observed by increasing the false alarm rate to as high as 10 times of the default false alarm rate parameter, i.e. $y' = 2.9\%$, while keeping the default detection rate $x = 83.33\%$. In this case the $U_D^{'}$ slightly decreases as false alarm rate increases while the $U_A^{'}$ remains constant. From both observations, it is obvious that the impact of varying detection rate x towards both players' utilities U_A and U_D is more sensitive than the variation of false alarm rate y.

6.7 Conclusion

In this chapter, a multi-stage Bayesian game model for IDS implementation in access control system of a M2M distributed local cloud framework is proposed. The feasible Nash equilibrium for each stage game is presented, which requires the defender to continuously update its belief towards its opponent. An analytical framework for a rational attacker and defender is provided, where the defender has a set of security assets to be protected that is also subject to the attack from the opponent, under a certain constraint of the attack and monitoring resource. The proposed model and analytical framework are evaluated numerically, and an interesting fact was found out that having to monitor a set of security assets helps the defender to improve its posterior belief update.

In this study, a defender's further action towards the attacker when its posterior belief reaches a certain level, e.g. kick the attacker out from the local cloud, has not been included in the game formulation. It is interesting to study the game when this action and another response action from the attacker, e.g. disassociate from the local cloud, are included, and to derive the best strategies for each player in such circumstances. Another interesting outlook is to study the game in which multiple attackers/defenders are involved, i.e. a coalition game.

6.8 References

[1] Tansu Alpcan and T. Basar. A game theoretic analysis of intrusion detection in access control systems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, 2004.

[2] M. Bloem, Tansu Alpcan, and T. Basar. Intrusion response as a resource allocation problem. In *Decision and Control, 2006 45th IEEE Conference on*, 2006.

[3] Lin Chen and J. Leneutre. A game theoretical framework on intrusion detection in heterogeneous networks. *Information Forensics and Security, IEEE Transactions on*, 4(2):165–178, 2009.

[4] Yi-Ming Chen, Dachrahn Wu, and Cheng-Kuang Wu. A game theoretic framework for multi-agent deployment in intrusion detection systems. In *Security Informatics*, volume 9 of *Annals of Information Systems*, pages 117–133. Springer US, 2010.

[5] Jason Crampton and Michael Huth. Towards an access-control framework for countering insider threats. In *Insider Threats in Cyber Security*, volume 49 of *Advances in Information Security*, pages 173–195. Springer US, 2010.

[6] ETSI. Machine-to-machine communications (m2m); functional architecture. Technical report, European Telecommunications Standards Institute (ETSI), 2011.

[7] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.

[8] IoT-A. Internet of things - architecture. *http://www.iot-a.eu*.

[9] George Labropoulos. D1.2.1 - user and system requirements. Technical report, BETaaS, 2012.

[10] Feng Li and Jie Wu. Hit and run: A bayesian game between malicious and regular nodes in manets. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, 2008.

[11] Yu Liu, Cristina Comaniciu, and Hong Man. A bayesian game approach for intrusion detection in wireless ad hoc networks. In *Proceeding from the 2006 workshop on Game theory for communications and networks*, GameNets '06. ACM, 2006.

7 Conclusions

This dissertation presents some challenges in the field of access control in IoT/M2M Cloud platform and gives solutions for them. There are three main topics that are discussed in this dissertation: generic access control model for IoT/M2M networks; access control and mobility management in RFID middleware; intrusion detection in access control system for M2M local cloud platform. In this chapter, the dissertation has been concluded and future direction of for the access control in IoT/M2M Cloud platform is given.

7.1 Access control and mobility management in RFID middleware

The inter-enterprise RFID middleware sub-system generates highly dynamic and huge amounts of RFID events data that creates a big challenge in managing the access of such data. The overall RFID events data contains some business context information, such as business step, business transaction type, business location, disposition, etc, which may compromise the privacy of the organization if such information is exploited by the attacker, e.g. the attacker can analyse the production volume, sales activity, etc of a company. As far as the access control is concerned, a fine-grained access control, i.e. disclose only specific portions of information or fields of data, is required to tackle the

privacy issue. In addition, an efficient access control mechanism is equally important to deal with huge data that dynamically changes over time. The proposed solution to such challenge is presented in Chapter 2. The access policy in the proposed mechanism is designed to take the profile of the accessing entity and results in a suitable set of access rules of the corresponding entity to the object(s), thus the access policy can be dynamically reuse for any user that even had no relationship before. The access policy is defined in such a way that it can express the dynamic context information well, e.g. time span and EPC IDs range. A system implementation which supports inter-operability with the RFID middleware defined by EPCglobal standard, is performed to evaluate the proposed access control model. Finally, some qualitative comparison between the proposed access control model with the existing ones, namely X-GTRBAC and AAL, and it can be shown the proposed access control model complements the features that are not available in each of them in terms of fine-grained access to data, trust, and inter-operability. However, it would have been more interesting to perform a quantitative study comparison against the existing models, i.e. performance measurements of the existing models implementation, which would also give a more solid analysis. In any case, it has been shown that the hypotheses 1.6.1.1.2 is justified.

Besides the challenge in privacy and access control in particular, the location or mobility management of the RFID middleware sub-system infrastructure is not well defined by EPCglobal standard. Additionally, as IP is one of the strongest candidate in inter-connecting billions of devices and network technologies, including RFID, over the internet, an IP-based mobility management in RFID allows smooth integration with IP-based service. Currently, the ECPglobal utilizes an ONS, an entity based on DNS implementation, to resolve the network address of the EPCIS repository that stores the information about the EPC. However, when a tag (represented with its EPC ID) moves to another domain, e.g. company or organization, the EPCglobal left the tag mobility management to the system implementation. In Chapter 3, an efficient, scalable, and secure mobility management of the inter-enterprise RFID sub-system based on EPCglobal standard is proposed. The proposed solution comprises an integration of a SIP based mobility management in the overall inter-enterprise RFID sub-system architecture and a secure access control based on the one proposed in Chapter 2. The test bed implementation is done in order to evaluate the performance of the proposed solution. The performance evaluation results show that the proposed solution greatly reduced the delay and is easily scale to many number of tags as compared with one of the existing system used in the study.

7.2 Capability-based access control for IoT/M2M networks and cloud platform

As a solution to issues particular in access control for the IoT/M2M, such as scalability issues due to huge numbers of devices and users, and the dynamic nature of IoT/M2M, the Capability-based Context-Aware Access Control (CCAAC) is proposed and designed in Chapter 4. The proposed CCAAC model solves the scalability issue in the IoT/M2M by using the capability – sometimes referred to as token – owned by the requesting entity, as means of authorizing an entity to access the protected resource. In order to solve the dynamic nature of distributed system such as IoT/M2M, the proposed CCAAC incorporates the context into the capability that becomes part of the access evaluation along with the capability validity check. The model also provides an authority delegation framework which is necessary in a system to operate in a distributed manner and to deal with the dynamic nodes connectivity in the IoT/M2M network. The authority delegation performed through a secure capability propagation. The privacy issue which becomes one of the primary concern in adopting the IoT is also addressed in the proposed CCAAC by introducing a virtual identity. The virtual identity is linked with the access policies of an object or resource to be accessed where the object owner can decide how it is to be seen and what information to be disclosed. However, the formal modelling of the incorporated context in the capability, i.e. by defining the exact types, attributes, and properties of the context, is left open, which opens an opportunity for future extension. In addition, the trust which plays an important role especially in the access decision as well as delegation has not been considered in the proposed CCAAC. The privacy issue which becomes one of the primary concern in adopting the IoT is also addressed in the proposed CCAAC by introducing a virtual identity. The virtual identity is linked with the access policies of an object or resource to be accessed where the object owner can decide how it is to be seen and what information to be disclosed. Finally, the proposed CCAAC is evaluated in order to measure its resilience against some attacks, especially man-in-the-middle and replay attacks. The evaluation results show that by introducing a nonce and lightweight cryptographic operation, the proposed access control mechanism is resilience against the previously mentioned types of attacks, plus the authentication can be achieved as well. As a conclusion, the hypotheses 1.6.1.3.1 is confirmed through these arguments and evaluation results, although some additional features with respect to context and trust could have been done to complete the overall CCAAC model.

Another capability-based access control model has been designed and implemented in Chapter 5 as an extension of the CCAAC model for the IoT/M2M cloud platform. Various issues pertaining access control, especially in the IoT/M2M cloud platform, such as the scalability and enforcing least privilege access principle issues in ACL, RBAC, or ABAC, access delegation issue in the API keys and OAuth based access control system, and the issue of controlling capability propagation in the typical capability based access control, are addressed by the proposed access control method. Several uses cases have shown that the proposed capability based access control is suitable to the various operations of the IoT/M2M cloud platform. It should also be noted that, in principal the proposed access control model is also suitable for any kind of cloud computing platform.

7.3 Intrusion detection in access control system for M2M local cloud platform

The IoT/M2M local cloud platform comprises a collection of distributed IoT/M2M gateways cooperating to each other so as to provide a richer set of combined M2M services from different gateways. One of the challenging issue is how to protect a set of resources or security assets owned by each gateway by means of access control mechanism, in the presence of uncertainty about the maliciousness of the other gateway(s) within the local cloud system. One solution of this issue is to incorporate an intrusion detection system (IDS) with the access control system. In addition, each gateway may be subject to some resource constraint (e.g. power) that influence how much it can monitor its assets using the IDS as well as the attacking effort by the attacker, thus necessitates the analysis of the optimum monitoring and attacking efforts (i.e. probability) to help the optimum configuration in the platform. An analytical framework that models the interaction between two gateways, i.e. attacker and defender, with some resource constraint and based on a two-player multi-stage Bayesian game is proposed in Chapter 6. The existence of the Perfect Bayesian Equilibrium (PBE) is proved in the proposed analytical framework and a set of PBE strategies in each stage game is constructed. A set of numerical analysis has been done to validate the proposed framework. Some important findings in the numerical analysis can be summarized as follows: the mixed strategies equilibrium for both players in the proposed framework are consistent throughout the whole game stages, the belief update of the defender is improved thanks to multiple assets being monitored simultaneously, and the impact of varying the detection rate towards both players' utilities is more sensitive than the

variation of the false alarm rate of the IDS. These findings show that the proposed multi-stage Bayesian game can be used to optimally configure the IDS in the access control system of an M2M local cloud platform given some system parameters, e.g. detection and false alarm rate of the IDS, the number and security values of the assets, and the resource constraints. In conclusion, the hypotheses 1.6.1.1.4 is proven, although some necessary adaptation or tweaking in the parameters, especially the monitoring, attacking, and false alarm costs, will still be needed in order to apply this analytical framework in the real system.

7.4 Future directions

The research work on the CCAAC can be extended into several directions. The access control mechanism considered in this work does not include the actual authentication method, although some level of authentication can be achieved by employing a simple light-weight cryptographic method. This work can be extended by incorporating the CCAAC model with light-weight authentication supporting capability, such as an ECC based key management and authentication, and to evaluate it with a more realistic adversaries model. The authority delegation framework in the proposed CCAAC assumes trust relationships that are already established among entities in different federation domain. The future extension will consider the case in which no prior knowledge of the trust relationship between two network domains in Federated IoT. Unlike the approach used in this paper, an additional entity that is trusted by both domains, for instance Identity Provider (IdP), needs to be involved in the design. Finally, prototype implementation of all aspects in CCAAC is something to look forward as a future step.

The access control management in the inter-enterprise RFID system relies on the trusted third parties entity to obtain the user profile. However, the mechanism on dealing with the trust management has not been addressed yet. The proposed model can be extended by incorporating Public Key Infrastructure (PKI) Certificate Authority (CA), e.g. X.509 Certificate, into the access control framework. Accordingly, the future update in the system implementation to support such extension is required.

In the research study on the intrusion detection in access control system for M2M local cloud platform, a defender's further action towards the attacker when its posterior belief reaches a certain level, e.g. kick the attacker out from

the local cloud, has not been included in the game formulation. It is interesting to study the game when this action and another response action from the attacker, e.g. disassociate from the local cloud, are included, and to derive the best strategies for each player in such circumstances. Another interesting outlook is to study the game in which multiple attackers/defenders are involved, i.e. a coalition game. Lastly, the future step also includes the implementation of this analytical framework in the real M2M local cloud platform.

A

Publications

A.1 Journal paper

- **B. Anggorojati**, N.R. Prasad, R. Prasad, "Efficient Fine Grained Access Control for RFID Inter-Enterprise System," Journal of Cyber Security and Mobility, vol. 2, no. 3 & 4, pp. 221-242, 2013
- P.N. Mahalle, **B. Anggorojati**, N.R. Prasad, R. Prasad, "Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things," Journal of Cyber Security and Mobility, vol. 1, no. 4, pp. 309-348, 2013.

A.2 Book chapter

- **B. Anggorojati**, P.N. Mahalle, N.R. Prasad, R. Prasad, "Secure Access Control and Authority Delegation based on Capability and Context Awareness for Federated IoT," Internet of Things and M2M Communications. ed. / Fabrice Theoleyre; Ai-Chun Pang. Denmark : River Publisher, 2013. p. 135-160 (The River Publishers Series in Information Science and Technology).

A.3 Conference papers

A.3.1 First Author

- **B. Anggorojati**, N.R. Prasad, R. Prasad, "Secure Capability-based Access Control in the M2M Local Cloud Platform," Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2014 IEEE 4th International Conference on, May, 2014.
- **B. Anggorojati**, N.R. Prasad, R. Prasad, "An Intrusion Detection Game in Access Control System for the M2M Local Cloud Platform," the 19th Asia Pasific Conference on Communications (APCC) 2013, Bali, Indonesia, 2013.
- **B. Anggorojati**, P.N. Mahalle, N.R. Prasad, R. Prasad, "Efficient and Scalable Location and Mobility Management of EPCglobal RFID System," the 16th WPMC, Atalantic City, NJ, USA, 2013.
- **B. Anggorojati**, P.N. Mahalle, N.R. Prasad, R. Prasad, "Capability-based Access Control Delegation Model on the Federated IoT Network," the 15th WPMC, Taipei, Taiwan, 2012.
- **B. Anggorojati**, K. Çetin, A. Mihovska, N. R. Prasad, "RFID Added Value Sensing Capabilities: European Advances in Integrated RFID-WSN Middleware," Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on, pp.1-3, 21-25 June 2010.
- **B. Anggorojati**, K. Çetin, B. K. Çetin, N. R. Prasad, "An implementation study for creating a research platform for telehomecare and Easy Life", 1st Wireless Vitae conference, Aalborg, May, 2009.

A.3.2 Co-author

- P.N. Mahalle, **B. Anggorojati**, N.R. Prasad, R. Prasad, "Identity driven Capability based Access Control (ICAC) scheme for the Internet of Things," the 6th IEEE ANTS, Bengaluru, India, 2012.
- P.N. Mahalle, **B. Anggorojati**, N.R. Prasad, R. Prasad, "Identity Establishment and Capability Based Access Control (IECAC) Scheme for Internet of Things," the 15th WPMC, Taipei, Taiwan, 2012.

- M. V. Ramkumar, **B. Anggorojati**, A. L. Stefan, N.R. Prasad, R. Prasad, "QoS-Guaranteed Admission Control for OFDMA-based Systems," 2nd IEEE International Workshop on Management of Emerging Networks and Services (IEEE MENS 2010), in conjunction with IEEE Globecom 2010, December, 2010.
- A. Mihovska, **B. Anggorojati**, J. Luo, S. Kyriazakos, N. R. Prasad, "Multi-Stage Ad mission Control for Load Balancing in Next Generation Systems," 11th WPMC, September 2008.
- F. Meucci, A. Mihovska, **B. Anggorojati**, N. R. Prasad, "Joint Resource Allocation and Admission Control Mechanism for an OFDMA-Based System," 11th WPMC, September 2008.

A.4 Project deliverables

- "D4.8.3: Integration of cooperation on WINNER II System Concept," WINNER II Project Deliverable Report, November 2007.
- "D6.12.3: Report on Validation and Implementation of Key WINNER cooperation Functionalities," WINNER II Project Deliverable Report, November 2007.
- "ISISEMD User and System Requirements," ISISEMD Project Deliverable Report, July 2009.
- "ISISEMD Privacy and Security Requirements Analysis," ISISEMD Project Deliverable Report, September 2009.
- "D3.2.1: ISISEMD Integrated Prototype Evaluation Plan," ISISEMD Project Deliverable Report, August 2010.
- "D3.2.2: ISISEMD Pilot Operation and Evaluation," ISISEMD Project Deliverable Report, June 2011.
- J. Soldatos, N. Kefalakis, N. Leontiadis, N. Konstantinou, N. Mitton, L. Schmidt, R. Dagher, M. David, **Bayu Anggorojati**, S. Frattasi, N. Prasad, D. Donsez, G. Pedraza, K. Gama, "D3.4b: Core ASPIRE Middleware Infrastructure (Final Version)," ASPIRE Project Deliverable Report, June 2010.
- J. Soldatos, N. Kefalakis, N. Leontiadis, D. Donsez, G. Pedraza, K. Gama, J. Estublier, **Bayu Anggorojati**, S. Frattasi, N. Prasad, "D3.5: End-to-End Infrastructure Management," ASPIRE Project Deliverable Report, June 2011.

- J. Soldatos, N. Kefalakis, D. Donsez, G. Pedraza, L. Zhang, H. Moran, S. Yousuf, **Bayu Anggorojati**, "D4.5: Integrated Development Environment for RFID Development," ASPIRE Project Deliverable Report, June 2011.
- "D2.3: Technical system specification and integration plan," LIFE 2.0 Project Deliverable Report, April 2012.
- "D2.6: Service Customisation Report," LIFE 2.0 Project Deliverable Report, July 2012.
- "D3.2.2: BETaaS APIs," BETaaS Project Delivery Report, February 2014.
- "D4.1: BETaaS Platform - Initial release," BETaaS Project Delivery Report, October 2013.
- "D2.2.1: Specification of the extended capabilities of the platform," BETaaS Project Delivery Report, June 2013.
- "D2.1.1: Specification of the basic capabilities and content use of the platform," BETaaS Project Delivery Report, March 2013.
- "D1.4.1: TaaS Reference Model," BETaaS Project Delivery Report, March 2013.
- "D1.1: State of the Art Review," BETaaS Project Delivery Report, December 2012.